

Mathematics and Visualization

Series Editors

Gerald Farin
Hans-Christian Hege
David Hoffman
Christopher R. Johnson
Konrad Polthier
Martin Rumpf

VISIT...

LANZAROTE
Caliente.COM

Xue-Cheng Tai
Knut-Andreas Lie
Tony F. Chan
Stanley Osher

Editors

Image Processing Based on Partial Differential Equations

Proceedings of the International Conference
on PDE-Based Image Processing and Related
Inverse Problems, CMA, Oslo, August 8–12, 2005

With 174 Figures, 22 in Color and 18 Tables

 Springer

Xue-Cheng Tai
Professor of Mathematics
Department of Mathematics
University of Bergen,
Johannes Brunsgate 12,
Bergen, N-5008, Norway
tai@mi.uib.no

Knut-Andreas Lie
SINTEF ICT, Dept. Applied Math.
PO Box 124 Blindern
N-0314 Oslo, Norway
Knut-Andreas.Lie@sintef.no

Tony F. Chan
Assistant Director for Math & Physical
Sciences Directorate
The National Science Foundation
4201 Wilson Boulevard
Arlington, Virginia 22230
USA
chan@math.ucla.edu

Stanley Osher
Department of Mathematics
Math Science Building
University of California at Los Angeles
520 Portola Plaza
Los Angeles, CA 90095, USA
sjo@math.ucla.edu

Library of Congress Control Number: 2006935256

Mathematics Subject Classification (2000): 35-06, 49-06 (49L25, 49M15, 49M30, 49N45), 65-06 (65K10, 65M06, 65M32, 65M55), 76-06 (76S05)

ISBN-10 3-540-33266-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-33266-4 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com
© Springer-Verlag Berlin Heidelberg 2007

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting by the authors and SPi using a Springer L^AT_EX macro package
Cover design: *design & production* WMXDesign GmbH, Heidelberg

Printed on acid-free paper SPIN: 11693628 46/SPi/3100 5 4 3 2 1 0

Preface

The book contains twenty-two original scientific research articles that address the state-of-the-art in using partial differential equations for image and signal processing. The articles arose from presentations given at the international conference on *PDE-Based Image Processing and Related Inverse Problems*, held at the Centre of Mathematics for Applications, University of Oslo, Norway, August 8-12, 2005.

The purpose of the conference was to bring together international researchers to present various aspects of new developments in using numerical techniques for partial differential equations to analyse and process digital images. Various aspects of new trends and techniques in this field were discussed in the conference, covering the following topics:

- Level set methods and applications
- Total variation regularization and other nonlinear filters
- Noise analysis and removal
- Image inpainting
- Image dejittering
- Optical flow estimation
- Image segmentation
- Image registration
- Analysis and processing of MR images and brain mapping
- Image construction techniques
- Level set methods for inverse problems

Inverse problems for partial differential equations have large areas of applications. Although image analysis and PDE inverse problems seem to be unrelated at a first glance, there are many techniques used in one of these two areas that are useful for the other. One goal of the conference was to highlight some of the recent efforts in merging some of the techniques for these two research areas.

We have arranged the twenty-two research articles of the book in six parts

Part I Digital Image Inpainting, Image Dejittering, and Optical Flow Estimation

Part II Denoising and Total Variation Methods

Part III Image Segmentation

Part IV Fast Numerical Methods

Part V Image Registration

Part VI Inverse Problems

The book collects new developments in these topics and points to the newest literature results. As such, it should be a good resource for people working on related problems, as well as for people who are new in the field. The book should also be suitable for readers working with computer vision and visualization, image and signal processing, as well as medical imaging. Moreover, the partial differential equations used for different problems discussed herein provide some rich research topics for people working with mathematical analysis and numerical simulation.

To ensure the scientific quality of the contributions to this book, each contributed paper was carefully reviewed. Special thanks go to all contributors and referees, without whom making this book would not have been possible.

Finally, we wish to thank those who supported and helped to organize the conference. First and foremost it is a pleasure to acknowledge the generous financial support from the Centre of Mathematics for Applications (CMA) and in particular the great help offered by Helge Galdal who has contributed to the practical work in organising the conference. In addition, partial financial support was given by Centre of Integrated Petroleum Research (University of Bergen), Simula Research Laboratory, and the Research Council of Norway (grant number 169281/V30). Moreover, we would like to thank the organising committee: Helge Galdal, Knut-Andreas Lie, Arvid Lundervold, Marius Lysaker, Hans Munthe-Kaas, Xue-Cheng Tai, Ragnar Winther, and Sigurd Aanonsen, for valuable contributions for making the conference a successful one. The participants of the conference deserve special thanks for making the conference a memorable event. Last but not least, the friendly and effective collaboration with Springer-Verlag through Martin Peters and Ute McCrory is kindly appreciated.

Bergen/Oslo/Los Angeles,
August 2006

Xue-Cheng Tai
Knut-Andreas Lie
Tony F. Chan
Stanley Osher

Contents

Part I Digital Image Inpainting, Image Dejittering, and Optical Flow Estimation

Image Inpainting Using a TV-Stokes Equation <i>Xue-Cheng Tai, Stanley Osher, Randi Holm</i>	3
Error Analysis for H^1 Based Wavelet Interpolations <i>Tony F. Chan, Hao-Min Zhou, Tie Zhou</i>	23
Image Dejittering Based on Slicing Moments <i>Sung Ha Kang, Jianhong (Jackie) Shen</i>	35
CLG Method for Optical Flow Estimation Based on Gradient Constancy Assumption <i>Adam Rabcewicz</i>	57

Part II Denoising and Total Variation Methods

On Multigrids for Solving a Class of Improved Total Variation Based Staircasing Reduction Models <i>Joseph Savage, Ke Chen</i>	69
A Method for Total Variation-based Reconstruction of Noisy and Blurred Images <i>Qianshun Chang, Weicheng Wang, Jing Xu</i>	95
Minimization of an Edge-Preserving Regularization Functional by Conjugate Gradient Type Methods <i>Jian-Feng Cai, Raymond H. Chan, Benedetta Morini</i>	109
A Newton-type Total Variation Diminishing Flow <i>Wolfgang Ring</i>	123

Chromaticity Denoising using Solution to the Skorokhod Problem	
<i>Dariusz Borkowski</i>	149

Improved 3D Reconstruction of Interphase Chromosomes Based on Nonlinear Diffusion Filtering	
<i>Jan Hubený, Pavel Matula, Petr Matula, Michal Kozubek</i>	163

Part III Image Segmentation

Some Recent Developments in Variational Image Segmentation	
<i>Tony Chan, Mark Moelich, Berta Sandberg</i>	175

Application of Non-Convex BV Regularization for Image Segmentation	
<i>Klaus Frick, Otmar Scherzer</i>	211

Region-Based Variational Problems and Normal Alignment – Geometric Interpretation of Descent PDEs	
<i>Jan Erik Solem, Niels Chr. Overgaard</i>	229

Fast PCLSM with Newton Updating Algorithm	
<i>Xue-Cheng Tai, Chang-Hui Yao</i>	249

Part IV Fast Numerical Methods

Nonlinear Multilevel Schemes for Solving the Total Variation Image Minimization Problem	
<i>Tony F. Chan, Ke Chen, Xue-Cheng Tai</i>	265

Fast Implementation of Piecewise Constant Level Set Methods	
<i>Oddvar Christiansen, Xue-Cheng Tai</i>	289

The Multigrid Image Transform	
<i>Paul M. de Zeeuw</i>	309

Minimally Stochastic Schemes for Singular Diffusion Equations	
<i>Bernhard Burgeth, Joachim Weickert, Sibel Tari</i>	325

Part V Image Registration

Total Variation Based Image Registration
Claudia Frohn-Schauf, Stefan Henn, Lars Hömke, Kristian Witsch 343
**Variational Image Registration Allowing
for Discontinuities in the Displacement Field**
Sven Kabus, Astrid Franz, Bernd Fischer 363

Part VI Inverse Problems

**Shape Reconstruction from Two-Phase Incompressible Flow
Data using Level Sets**
Rossmay Villegas, Oliver Dorn, Miguel Moscoso, Manuel Kindelan 381
**Reservoir Description Using a Binary Level Set Approach
with Additional Prior Information About the Reservoir
Model**
*Lars Kristian Nielsen, Xue-Cheng Tai, Sigurd Ivar Aanonsen, Magne
S. Espedal* 403

Color Figures 427

**Digital Image Inpainting, Image Dejittering,
and Optical Flow Estimation**

Image Inpainting Using a TV-Stokes Equation

Xue-Cheng Tai¹, Stanley Osher², and Randi Holm¹

¹ Department of Mathematics, University of Bergen, Johs. Bruns gt. 12, N-5007 Bergen, Norway. E-mail: tai@mi.uib.no, url: <http://www.mi.uib.no/~tai>.

² Department of Mathematics, UCLA, California, USA. E-mail: sjo@math.ucla.edu

Summary. Based on some geometrical considerations, we propose a two-step method to do digital image inpainting. In the first step, we try to propagate the isophote directions into the inpainting domain. An energy minimization model combined with the zero divergence condition is used to get a nonlinear Stokes equation. Once the isophote directions are constructed, an image is restored to fit the constructed directions. Both steps reduce to the solving of some nonlinear partial differential equations. Details about the discretization and implementation are explained. The algorithms have been intensively tested on synthetic and real images. The advantages of the proposed methods are demonstrated by these experiments.

1 Introduction

For a digital image, inpainting refers to the process of filling-in missing data. It ranges from removing objects from an image to repairing damaged images and photographs.

The term of “digital inpainting” seems to have been introduced into image processing by Bertalmio, Sapiro, Caselles and Ballester [2]. In the past few years, several different approaches have been proposed to tackle this complicated image processing task. The basic idea for most of the inpainting techniques is to do a smooth propagation of the information in the region surrounding the inpainting area and interpolating level curves in a proper way [2, 21, 6]. However, there are different strategies to achieve these goals. In [2], the authors proposed to minimize an energy to compute the restored image and this results in the solving of coupled nonlinear differential equations. In a related work [4], this idea was further extended to guarantee that the level curves are propagated into the inpainting domain. In [3], a connection between the isophote direction of the image and the Navier-Stokes equation was observed and they proposed to solve transport equations to fill in the inpainting domain. This is related to our method. Another related work is [11] where a minimization of the divergence is done to construct optical flow functions.

The work of [9, 7] minimizes the TV-norm of the reconstructed image to fill in the missing data. In later work [8, 10], energy involving the curvature of the level curves is used and this is in some sense trying to guarantee that the level curves are connected in a smooth fashion. The equations obtained from such models are highly nonlinear and of higher (fourth) order.

Recently, texture inpainting has attracted attention. In [5], the image in the surrounding area is first decomposed into texture and structure and then propagated into the inpainting domain in different ways. This idea to decompose texture and structure is also used in [12]. Some statistical approaches are used in [1] to do texture synthesis and structure propagation.

We may also mention some recent works which related the phase-field model and Ginzburg-Landau equation to image processing, [15, 16, 13, 12]. These ideas were used in [15, 16, 13] for image segmentation. In [12] they were used for image inpainting.

The idea used in this work was motivated by [19, 20, 2, 3]. We still follow the basic ideas of image inpainting, i.e., we are trying to propagate the information into the inpainting domain along the isophote directions. However, we choose a two-step method to carry out this task as in [20]. The first step involves trying to reconstruct the isophote directions for the missing data. The second step tries to construct an image fitting the restored directions. This is the same idea used in [20] to remove noise from digital images. One new idea which is essential to the present method is that we impose the zero divergence condition on the constructed directions. This guarantees that there exists an image such that its isophote directions are the restored vectors. This is important when the inpainting region is relatively large. In contrast to [3], we obtain our TV-Stokes equation from this consideration which implies that the obtained vectors have the smallest TV-norm. The solution of the Stokes equation will generally not have such a property. We also propose some novel ideas to modify the boundary condition for the inpainting domain to select the information that is propagated into the region. We have only tested our algorithms on propagated structure information. It is possible to combine it with texture inpainting as in [5].

This work is organized as follows. In section 2, we explain the detailed mathematical principles for our methods. First, some geometrical motivation is presented. These geometrical observations are then combined with energy minimization models to get the nonlinear equations which give our inpainting methods. Discretization and implementation details are then supplied. When solving the equations, it is rather easy to change the boundary conditions. Due to this flexibility, we show that it is rather easy to block some information from propagating into the inpainting region. Numerical experiments on real and synthetic images are supplied in Section 3 and comparisons with other methods are discussed.

2 The Mathematical Principles

Suppose that an image $u_0 : R \mapsto [a, b]$ is defined on a rectangle domain R . We shall assume that $\Omega \subset R$ is the domain where the data is missing. We want to fill in the information on Ω based in the geometrical and photometric information surrounding the region Ω . As in [2], we shall use information in a band B around the domain Ω . We shall use $\tilde{\Omega} = \Omega \cup B$ in the following.

2.1 Connection Between Digital Images and Flow Fields

In [3], the connection between image inpainting and fluid dynamics is done by observing that the isophote directions of an image correspond to an incompressible velocity field. This same observation will be used here in our work. However, the equation we shall use for the inpainting is different and is related to the work of [20]. We give a brief outline of the idea of [20] in the following.

Given scalar functions u and v , denote:

$$\nabla u = (u_x, u_y), \quad \nabla^\perp u = (-u_y, u_x), \quad \nabla \times (u, v) = u_y - v_x, \quad \nabla \cdot (u, v) = u_x + v_y.$$

Given an image d_0 , the level curves:

$$\Gamma(c) = \{x : d_0(x) = c, \quad \forall c \in (-\infty, \infty)\}.$$

have normal vectors $\mathbf{n}(x)$ and tangential vectors $\boldsymbol{\tau}(x)$ given by

$$\mathbf{n}(x) = \nabla d_0(x) \quad \boldsymbol{\tau}(x) = \nabla^\perp d_0(x).$$

The vector fields $\mathbf{n}$ and $\boldsymbol{\tau}$ satisfy

$$\nabla \times \mathbf{n}(x) = 0, \quad \nabla \cdot \boldsymbol{\tau}(x) = 0. \quad (1)$$

Suppose that the surface $d_0(x)$ is exposed to rain, then the rain will flow down the surface along the directions $-\mathbf{n}(x)$. One observation is that the surface d_0 can be constructed from the vector fields $\mathbf{n}(x)$ or $\boldsymbol{\tau}(x)$.

For image inpainting, the information of d_0 in the surrounding band B is known. Thus, we also know the normal and tangential vectors of d_0 in B . The main idea to fill in the information in Ω is to propagate the vector field $\mathbf{n}$ or $\boldsymbol{\tau}$ into the interior region Ω . Afterwards, we construct an image in region Ω to fit the computed vectors in Ω .

Define $\boldsymbol{\tau}_0 = \nabla^\perp d_0$. There are many different ways to propagate the vectors from B into Ω . In [3], incompressible, inviscid Euler equations are used. Here, we shall use an energy minimization model to propagate the vector fields, i.e., we shall solve

$$\min_{\nabla \cdot \boldsymbol{\tau} = 0} \int_{\tilde{\Omega}} |\nabla \boldsymbol{\tau}| dx + \frac{1}{\epsilon} \int_B |\boldsymbol{\tau} - \boldsymbol{\tau}_0|^2 dx \quad (2)$$

Above, $\int_{\tilde{\Omega}} |\nabla \boldsymbol{\tau}| dx$ is the total variation for vector field $\boldsymbol{\tau}$. We require $\nabla \cdot \boldsymbol{\tau} = 0$ to guarantee that the reconstructed vector field $\boldsymbol{\tau}$ is a tangential vector for the

level curves of a scalar function in the region $\tilde{\Omega}$. The penalization parameter ϵ is chosen to be very small to guarantee that $\boldsymbol{\tau} \approx \boldsymbol{\tau}_0$ in B . For most of the cases we have tested, it is enough to take B to be just one pixel wide around Ω . For such a case, we can take $\epsilon \rightarrow 0$ and thus the minimization problem reduces to find a $\boldsymbol{\tau}$ such that $\boldsymbol{\tau} = \boldsymbol{\tau}_0$ on $\partial\Omega$ which solves:

$$\min_{\nabla \cdot \boldsymbol{\tau} = 0} \int_{\Omega} |\nabla \boldsymbol{\tau}| dx. \quad (3)$$

We use the total variation norm of $\boldsymbol{\tau}$ (as usual in this subject) because the boundary value $\boldsymbol{\tau}_0$ may have discontinuities. In order to propagate such a discontinuity into the region Ω , we need to allow $\boldsymbol{\tau}$ to have discontinuities and thus the TV-norm is preferred to e.g., the H^1 -norm.

We use χ_B to denote the characteristic function over the domain B , i.e., $\chi_B = 1$ in B and $\chi_B = 0$ elsewhere. If we use a Lagrange multiplier λ to deal with the divergence constraint $\nabla \cdot \boldsymbol{\tau} = 0$, the Euler-Lagrange equation of (2) is:

$$\begin{cases} -\nabla \cdot \left(\frac{\nabla \boldsymbol{\tau}}{|\nabla \boldsymbol{\tau}|} \right) + \frac{\chi_B}{\epsilon} (\boldsymbol{\tau} - \boldsymbol{\tau}_0) - \nabla \lambda = 0 & \text{in } \tilde{\Omega}, \\ \nabla \cdot \boldsymbol{\tau} = 0 & \text{in } \tilde{\Omega}, \\ \nabla \boldsymbol{\tau} \cdot \boldsymbol{\nu} = \mathbf{0} & \text{on } \partial\tilde{\Omega}. \end{cases} \quad (4)$$

Here, $\boldsymbol{\nu}$ denotes the outer unit normal vector of $\partial\tilde{\Omega}$. Similarly, the Euler-Lagrange equation of (3) is:

$$\begin{cases} -\nabla \cdot \left(\frac{\nabla \boldsymbol{\tau}}{|\nabla \boldsymbol{\tau}|} \right) - \nabla \lambda = 0 & \text{in } \Omega, \\ \nabla \cdot \boldsymbol{\tau} = 0 & \text{in } \Omega, \\ \boldsymbol{\tau} = \boldsymbol{\tau}_0 & \text{on } \partial\Omega. \end{cases} \quad (5)$$

Once the tangential vector field $\boldsymbol{\tau}$ is available in $\tilde{\Omega}$, it is easy to obtain the normal vector field $\mathbf{n}$. Let u and v be the two components of the vector field $\boldsymbol{\tau}$, i.e., $\boldsymbol{\tau} = (u, v)$. Then, we have

$$\mathbf{n}(x) = \boldsymbol{\tau}^\perp(x) = (-v, u). \quad (6)$$

From the vector field $\mathbf{n}(x)$, we use the same idea as in [20, 2] to construct an image d whose normal vectors shall fit the computed vectors $\mathbf{n}(x)$. This is achieved by solving the following minimization problem:

$$\min \int_{\tilde{\Omega}} |\nabla d| - \nabla d \cdot \frac{\mathbf{n}}{|\mathbf{n}|} dx + \frac{1}{\epsilon} \int_B |d - d_0|^2 dx. \quad (7)$$

The penalization parameter ϵ can be chosen to be same as in (2). Or it can be chosen to be different. In case that B is only one pixel wide around Ω , the above minimization problem reduces to the following problem if we take $\epsilon \rightarrow 0$:

$$\min_d \int_{\Omega} |\nabla d| - \nabla d \cdot \frac{\mathbf{n}}{|\mathbf{n}|} dx \quad \text{and } d = d_0 \text{ on } \partial\Omega. \quad (8)$$

The Euler-Lagrange equation of (7) is:

$$\begin{cases} -\nabla \cdot \left(\frac{\nabla d}{|\nabla d|} - \frac{\mathbf{n}}{|\mathbf{n}|} \right) + \frac{\chi_B}{\epsilon} (d - d_0) = 0 \text{ in } \tilde{\Omega}, \\ \left(\frac{\nabla \boldsymbol{\tau}}{|\nabla \boldsymbol{\tau}|} - \frac{\mathbf{n}}{|\mathbf{n}|} \right) \cdot \boldsymbol{\nu} = \mathbf{0} \text{ on } \partial\tilde{\Omega}. \end{cases} \quad (9)$$

Similarly, the Euler-Lagrange equation of (8) is:

$$\begin{cases} -\nabla \cdot \left(\frac{\nabla d}{|\nabla d|} - \frac{\mathbf{n}}{|\mathbf{n}|} \right) = 0 \text{ in } \Omega, \\ d = d_0 \text{ on } \partial\Omega. \end{cases} \quad (10)$$

2.2 Discretization

We now explain some of the details in discretizing the equations derived in the last section for numerical simulations. For clarity, we shall only outline the details for algorithms (5) and (10). The discretization for (4) and (9) can be done in a similar way.

For simplicity, the gradient descent method will be used in our simulations. The gradient flow equation for $\boldsymbol{\tau}$ is:

$$\frac{\partial \boldsymbol{\tau}}{\partial t} - \nabla \cdot \left(\frac{\nabla \boldsymbol{\tau}}{\|\nabla \boldsymbol{\tau}\|} \right) - \nabla \lambda = 0 \quad \text{in } \Omega, \quad (11)$$

$$\nabla \cdot \boldsymbol{\tau} = 0, \text{ in } \Omega, \quad \boldsymbol{\tau} = \boldsymbol{\tau}_0 \quad \text{on } \partial\Omega. \quad (12)$$

where $\|\nabla \boldsymbol{\tau}\| = \sqrt{|u_x|^2 + |u_y|^2 + |v_x|^2 + |v_y|^2}$. We have tried two algorithms to solve (11)-(12). The first algorithm uses the following iterative procedure to update $\boldsymbol{\tau}$ and λ with the time step Δt_1 and initial values properly chosen:

$$\boldsymbol{\tau}^{n+1} = \boldsymbol{\tau}^n + \Delta t_1 \left[\nabla \cdot \left(\frac{\nabla \boldsymbol{\tau}^n}{\|\nabla \boldsymbol{\tau}^n\|} \right) + \nabla \lambda^n \right], \quad (13)$$

$$\lambda^{n+1} = \lambda^n + \Delta t_1 \nabla \cdot \boldsymbol{\tau}^n \quad (14)$$

The second algorithm updates $\boldsymbol{\tau}$ and λ by:

$$\boldsymbol{\tau}^{n+1} = \boldsymbol{\tau}^n + \Delta t_1 \left[\nabla \cdot \left(\frac{\nabla \boldsymbol{\tau}^n}{\|\nabla \boldsymbol{\tau}^n\|} \right) + \nabla \lambda^n \right], \quad (15)$$

$$-\Delta \lambda^{n+1} = \nabla \cdot \left[\nabla \cdot \left(\frac{\nabla \boldsymbol{\tau}^n}{\|\nabla \boldsymbol{\tau}^n\|} \right) \right]. \quad (16)$$

In (16), Δ denotes the Laplace operator and we impose a zero Neumann boundary condition for λ^{n+1} . If $\nabla \cdot \boldsymbol{\tau}^0 = 0$ and (16) is satisfied by all λ^n , then we see from (15) that

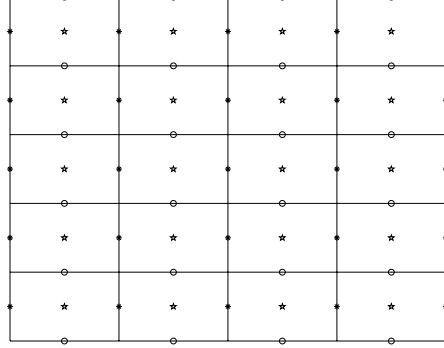


Fig. 1. The pixels and the approximation points for u, v, λ and d . The approximation points are: $*$ for u , $\circ$ for v , $\star$ for λ .

$$\nabla \cdot \tau^{n+1} = 0, \quad \forall n.$$

We use a staggered grid to approximate u, v and λ . Note that $\tau = (u, v)$ is used to construct d . When we try to compute d from (9) or (10), we are trying to enforce the following relation approximately: $u = -d_y$, $v = d_x$. Due to this relation, the grid points used in the approximation for u are chosen to be the points marked with $*$, see Figure 1. The approximation points for v are marked with $\circ$. The centers of the rectangle elements marked with $\star$ are used as the approximation points for λ . The vertices of the rectangular mesh are used as the approximation points for d . The horizontal axis represents the x -variable and the vertical axis represents the y -variable, c.f Figure 1.

For a given domain Ω , we use $U_h(\Omega)$ to denote all the approximation points $*$ for u inside Ω , $V_h(\Omega)$ to denote all the approximation points $\circ$ for v inside Ω , $\Lambda_h(\Omega)$ to denote all the approximation points $\star$ for λ inside Ω and $D_h(\Omega)$ to denote all the approximation points for d inside Ω . The updating formulae for (u, v) and λ for (13)-(14) are:

$$u^{n+1} = u^n + \Delta t_1 \left(D_x^- \left(\frac{D_x^+ u^n}{T_1^n} \right) + D_y^- \left(\frac{D_y^+ u^n}{T_2^n} \right) + C_x^{h/2} \lambda^n \right) \quad \text{on } U_h(\Omega), \quad (17)$$

$$v^{n+1} = v^n + \Delta t_1 \left(D_x^- \left(\frac{D_x^+ v^n}{T_2^n} \right) + D_y^- \left(\frac{D_y^+ v^n}{T_1^n} \right) + C_y^{h/2} \lambda^n \right) \quad \text{on } V_h(\Omega), \quad (18)$$

$$\lambda^{n+1} = \lambda^n + \Delta t_1 (C_x^{h/2} u^{n+1} + C_y^{h/2} v^{n+1}) \quad \text{on } \Lambda_h(\Omega). \quad (19)$$

Above, $D_x^\pm, D_y^\pm$ are the standard forward/backward finite difference operators and $C_x^{h/2}, C_y^{h/2}$ are the central finite difference operators with mesh size $h/2$.

h denotes the mesh size for the approximations and is taken to be one. The terms T_1^n and T_2^n are evaluated as in the following:

$$T_1^n = \sqrt{|D_x^+ u|^2 + |C_y^h u|^2 + |C_y^h v|^2 + |D_y^+ v|^2 + \epsilon} \text{ on } \Lambda_h(\Omega), \quad (20)$$

$$T_2^n = \sqrt{|C_x^h u|^2 + |D_y^+ u|^2 + |D_y^+ v|^2 + |C_y^h v|^2 + \epsilon} \text{ on } D_h(\Omega). \quad (21)$$

If we use the second algorithm to compute (u, v) and λ from (15)-(16), the solution of (16) is not unique due to the use of the Neumann boundary condition. We fix the value of λ to be zero at one point on the boundary to overcome this problem, which is standard for this kind of problem. Fast methods, like the FFT (Fast Fourier Transformation), can be used to solve (16).

Once the iterations for u and v have converged to a steady state, we use them to obtain d . Note that the relation between (u, v) and $\mathbf{n}$ is as in (6). Similar as in [20], the following gradient flow scheme is used to update d of (10):

$$\begin{aligned} d^{n+1} = d^n + \Delta t_2 \Bigg(& \left(D_x^- \left(\frac{D_x^+ d^n}{D_1^n} + \frac{v}{\sqrt{\hat{u}^2 + v^2 + \epsilon}} \right) \right. \\ & \left. + \left(D_y^- \left(\frac{D_y^+ d^n}{D_2^n} - \frac{u}{\sqrt{u^2 + \hat{v}^2 + \epsilon}} \right) \right) \right) \text{ on } D_h(\Omega). \end{aligned} \quad (22)$$

In the above, $\hat{u}, \hat{v}$ are the average values of the four nearest approximation points and

$$D_1^n = \sqrt{|D_x^+ d^n|^2 + |C_y^h d^n|^2 + \epsilon} \text{ on } D_h(\Omega), \quad (23)$$

$$D_2^n = \sqrt{|C_x^h d^n|^2 + |D_y^+ d^n|^2 + \epsilon} \text{ on } D_h(\Omega). \quad (24)$$

This iteration is the standard gradient updating for d . We could use the AOS scheme of [17, 18] to accelerate the convergence. The AOS scheme was first proposed in [17, 18]. It was later rediscovered in [22, 14] and used for image processing problems.

Up to now we have only explained the approximation details for (5) and (10). It is easy to see that the discretization for (4) and (9) can be done in a similar way. The Dirichlet or Neumann boundary conditions for the different equations are implemented in the standard way and we will omit the details.

2.3 Other Kind of Boundary Conditions

We have proposed two alternatives to deal with the information which is in the surrounding area of Ω , i.e.

- Using information in a narrow band around the inpainting region Ω and trying to propagate this information into the region Ω using equations (4) and (9).

- Using information of the two nearest pixels around the inpainting region Ω and using equations (5) and (10) to propagate the information into the region Ω .

There is no strong evidence about which of these two alternatives is better. In fact, numerical experiments show that this is image dependent. In most of the tests given in this work, we have used the boundary conditions (5) and (10).

In the following, we shall even propose another boundary condition to treat some special situations. For some images, we may want some of the information from the surrounding area to be propagated into Ω , while some other information from the surrounding area is not welcome to be so propagated, see Figures 9, 11, and 12. In order to deal with this kind of situation, we propose the following alternative:

- Decompose the boundary $\partial\Omega$ into two parts, i.e., $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$. For equation (5), replace the boundary condition by

$$a) \ \boldsymbol{\tau} = \boldsymbol{\tau}_0 \text{ on } \partial\Omega_D, \quad b) \ \boldsymbol{\tau} = \mathbf{0} \text{ on } \partial\Omega_N, \quad (25)$$

and replace the boundary condition of (10) by

$$a) \ d = d_0 \text{ on } \partial\Omega_D \quad b) \ \frac{\partial d}{\partial \boldsymbol{\nu}} = 0 \text{ on } \partial\Omega_N. \quad (26)$$

Condition (26.b) means that we do not want to propagate any information through $\partial\Omega_N$. Due to the fact that $\nabla d^\perp \approx \boldsymbol{\tau}$, condition (26.b) implies that we must have condition (25.b) for $\boldsymbol{\tau}$ on $\partial\Omega_N$. A similar procedure can be performed for equations (4) and (9).

3 Numerical Experiments

First, we explain how to choose ε , Δt_1 and Δt_2 in numerical implementations. We add ε to the denominator to avoid dividing by zero in (20)-(21) and (23)-(24). If ε is chosen to be large, the computed image will be smoothed a bit. If ε is chosen to be too small, it may slow down the convergence. We have chosen ε to be the same in (20)-(21) and (23)-(24), but it will differ from example to example.

With large Δt_1 and Δt_2 , the iterations will converge faster, but if they are too large, the scheme is unstable. For most experiments $\Delta t_1 \approx 0.03$ will lead to convergence of the normal vectors. A smaller Δt_1 will also work, but more iterations might be necessary. If the normal vectors are smooth, Δt_2 is less sensitive and can be chosen to be large. If the vector field is less smooth, Δt_2 must be smaller.

Example 1

In this example we test out our method on an image from a Norwegian newspaper. The image shows a man jumping from Jin Mao Tower, a 421 meter tall building in the city of Shanghai, see Figure 2. We want to remove the man and restore the city in the background.

The first part of the code computes the normal vectors in the missing region. From Figure 3 we see that the vectors are propagating into the inpainting region in a smooth fashion. When $\Delta t_1 = 0.03$ and $\epsilon = 10$ are used, a steady state is reached after 3000 iterations using (13)-(14). If we use (15)-(16), less than 1000 iterations are needed to reach a steady state, see Figure 3 e) and Figure 3 f).

The second part reconstructs the image using the computed normal vectors. Figure 4 shows how the man is gradually disappearing during the iterations. With $\Delta t_2 = 0.15$ it takes 30000 iterations before a steady state is reached. In the resulting image the man has disappeared completely and the background is restored in a natural way. There are no discontinuities in the sky, and the skyline is almost a straight line. It is nearly impossible to detect that the sky and the skyline contains the missing region.



Fig. 2. The original image. (Color image in Figure A.1.)

Example 2

We test our method on some well-known examples which have been tested by others using different methods [2]. We use these results to show the quality of the restored images compared with other methods.

In the example shown in Figure 5, red text is written over the picture. The text is the inpainting area, and we want to fill it with information from the image. With $\epsilon = 1$ and $\Delta t_1 = 0.03$ the normal vectors converge after 7000

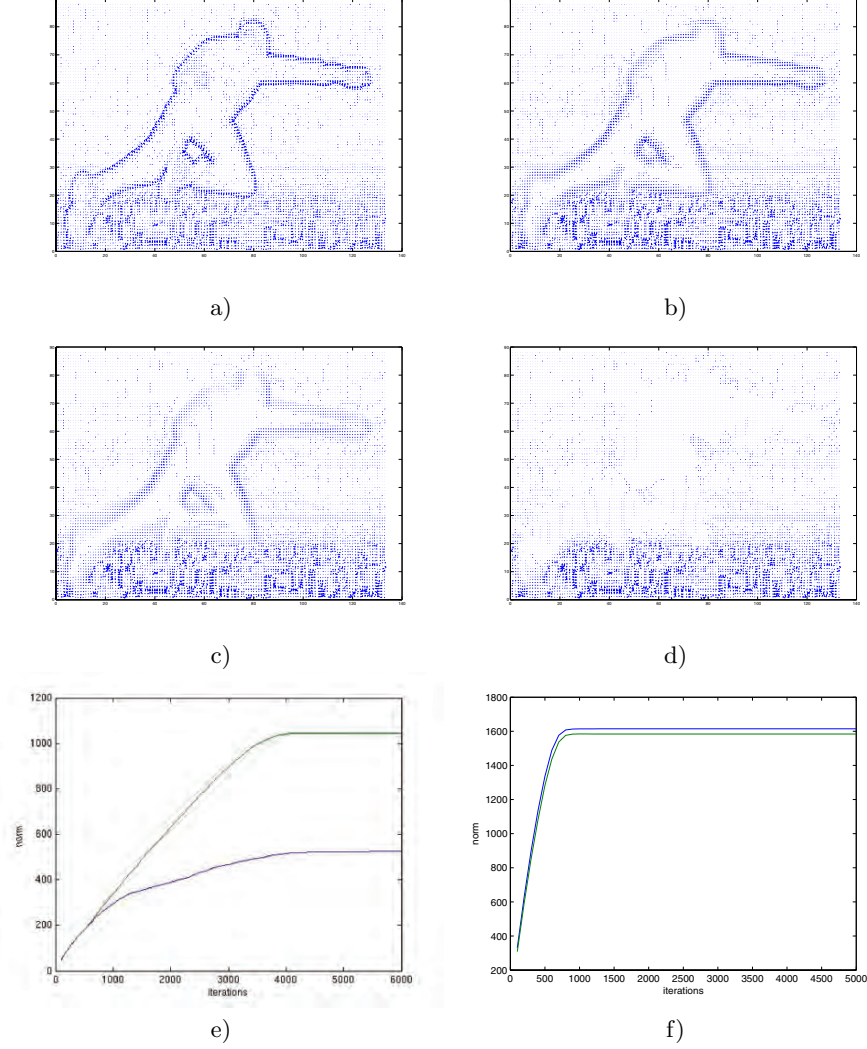


Fig. 3. The restored flow vector τ using (13)-(14) at different iterations. a) at iteration 0; b) at iteration 1000; c) at iteration 2000; d) at iteration 3000; e) The plot for $\|u\|$ and $\|v\|$ which shows that the equations (13)-(14) reach a steady state, i.e. at iteration 3000. f) In this plot, we show the convergence for $\|u\|$ and $\|v\|$ using equations (15)-(16). They reach steady states quicker than (13)-(14), i.e., at iteration 1000.

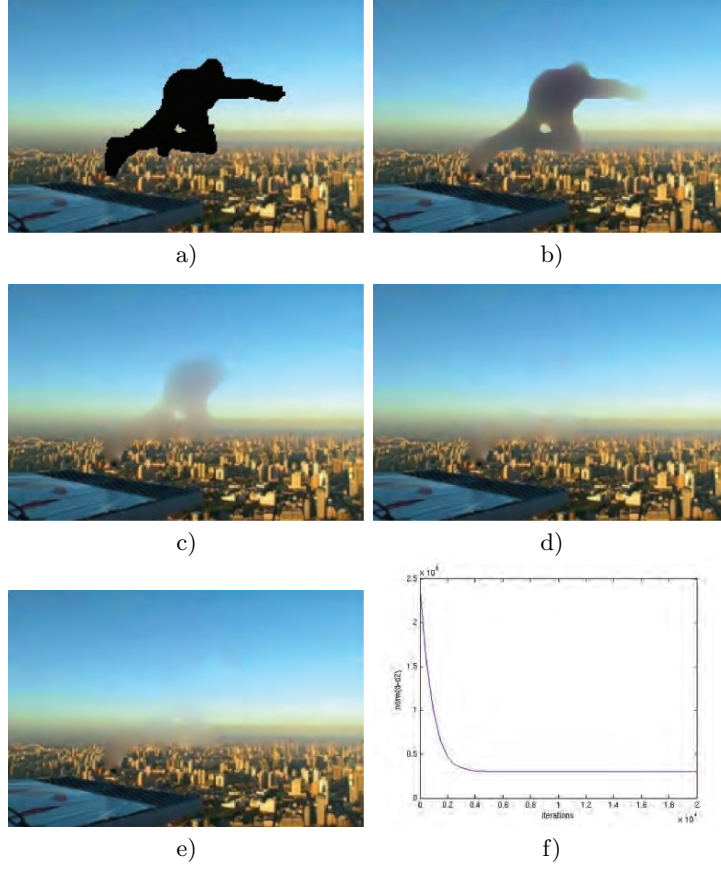


Fig. 4. The restored image d using equation (10) at different iterations. a) at iteration 0; b) at iteration 10000; c) at iteration 20000; d) at iteration 30000; e) The restored image using the new method (15)-(16) to find τ . f) The plot for $\|d - d_0\|$ which shows that the equation (5) reaches a steady state, i.e., at iteration 30000. f) The plot for $\|\tau^{n+1} - \tau^n\|$ which goes to zero very quickly which also shows the steady state is quickly reached. (Color images in Figure A.2.)

iterations for (13)-(14). The second part of the code converged after only 3000 iterations with $\Delta t_2 = 0.5$.

In Figure 6, another image which has been tested in the literature, is used here to compare our method with the others, [2, 1]. The image has the white text ‘Japanese animation’, and we want to remove this. An area around the text is lighter than the background and has to be restored as well. Figure 6 b) shows the manually obtained inpainting region. Figure 6 c) shows restored image. The values for Δt_1 and Δt_2 are chosen to be the same as in the previous example, and the convergence is nearly the same.



Fig. 5. a) The original image. b) The restored image using equations (5) and (10). c) The difference image. (Color images in Figure A.3.)

Figure 7 a) shows an old photo which has been damaged. We mark the inpainting region in white colour, as shown in Figure 7 b) and try to restore it. The result is shown in Figure 7 c).

The image in Figure 8 a) shows another situation where our algorithm can be applied. The image has a piece of musical notes written on it. A large amount of information is lost, but it is scattered on the image in narrow areas. The first part converges after 2500 iterations and the second part converges after 1000 iterations when using our algorithm for this image. The restored image in Figure 8 b) looks rather good.

Example 3

To test the code for the new boundary condition (25)-(26), we created a simple image, see Figure 9. Information is missing in a rectangle in the middle of the image which only has two intensity values. If we use Dirichlet boundary conditions (5)-(10), all information from the surrounding area will be transported into the inpainting region. If the Neumann boundary is used (25)-(26), it is possible to choose which intensity value to be selected to propagate into the

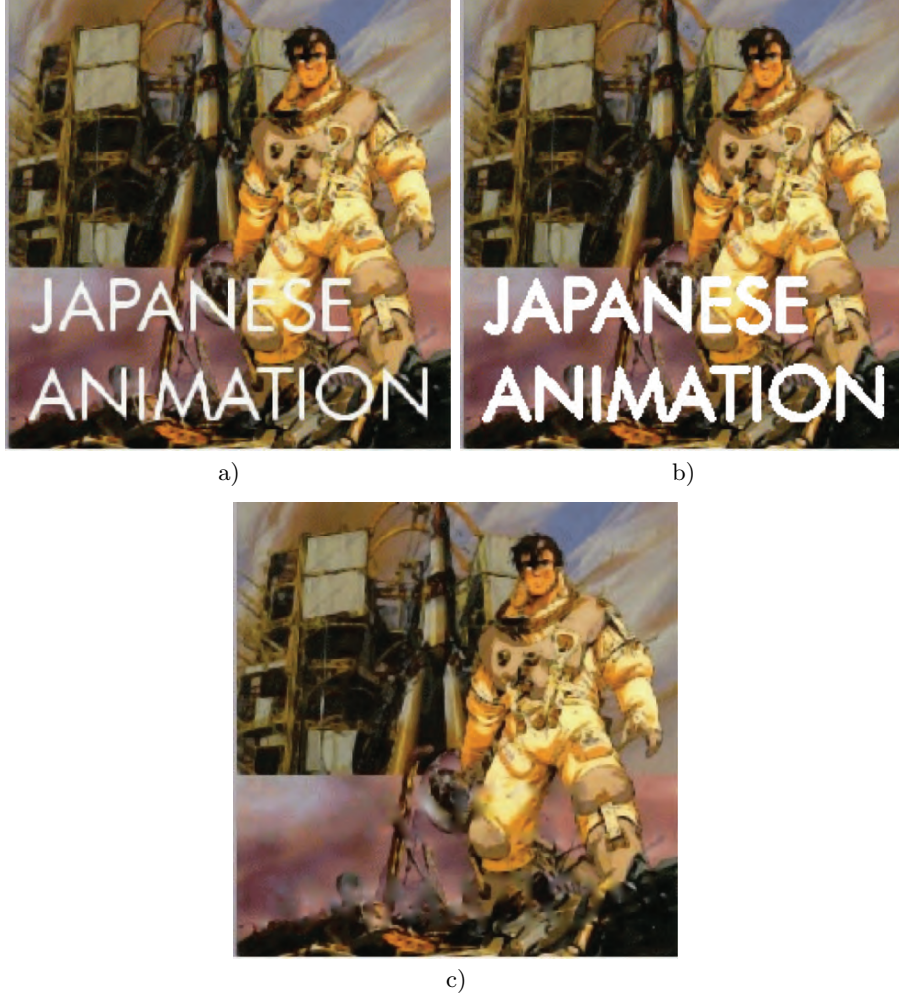


Fig. 6. a) The original image. b) The image with the inpainting region obtained manually. c) The restored image using equations (5) and (10). (Color images in Figure A.4.)

inpainting region. The result is shown in Figure 9. The result using Dirichlet boundary conditions is displayed in Figure 9 b). With $\epsilon=0.0001$, $\Delta t_1 = 0.01$, the normal vectors converged after 12000 iterations and with $\Delta t_2 = 0.2$ the second part converged after 25000 iterations. With a larger ϵ , the corners and the boundary close to the corners may be smeared.

Figure 9 c) shows a similar test with Dirichlet conditions on the upper half and with Neumann boundary conditions on the lower half of the boundary of

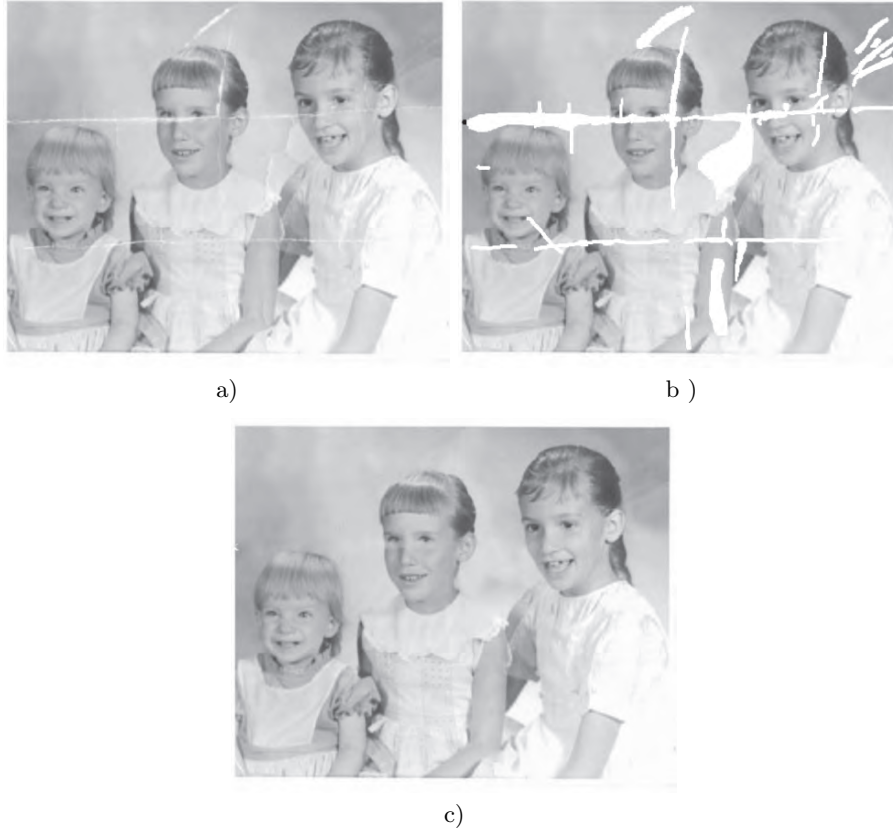


Fig. 7. a) The original image d_0 . b) The image with the inpainting region white. c) The restored image d .

the inpainting region. From Figure 9 c) we see that only one of the colours was selected and propagated to the interior.

Example 4

In this example, we process an image from the match between Norway and Croatia in the XIX Men's World Championship. We want to remove the Croatian player in Figure 10. When a Dirichlet condition is used around the whole boundary, Figure 11 a), colours from the Norwegian players propagate into the background. To make it look natural, it is necessary to use Neumann boundary conditions around the two Norwegian players. The inpainting region and the Neumann boundary are marked in Figure 11 b). Figure 11 c) shows the restored image using this new boundary condition. When Neumann boundary



a)



b)

Fig. 8. a) The image with the inpainting region white. b) The restored image using equations (5) and (10). (Color images in Figure A.5.)

condition is used, the colour on the Neumann boundary does not influence the interior.

Example 5

This example has more texture in the background. We want to remove the snowboarder and fill in the missing region. It is not desirable that the yellow object in the front propagates into the inpainting region. Figure 12 d) shows

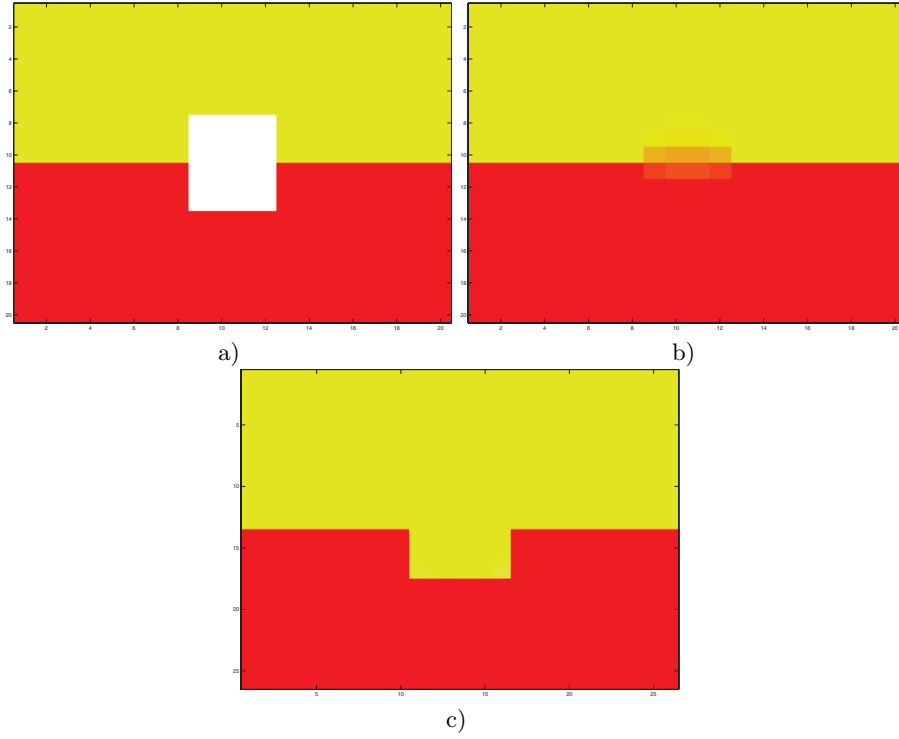


Fig. 9. a) The image with the inpainting region marked. b) The image obtained with Dirichlet boundary. c) The image obtained using Dirichlet and Neumann boundary conditions.



Fig. 10. An image from the match between Norway and Croatia in the XIX Men's World Championship. (Color images in Figure A.6.)



a)



b)



c)

Fig. 11. a) The restored image using Dirichlet boundary conditions. b) The image with the inpainting region shaded in gray. c) The restored image using Dirichlet and Neumann boundary conditions. (Color images in Figure A.7.)

that the best result is obtained with Neumann conditions on part of the boundary.

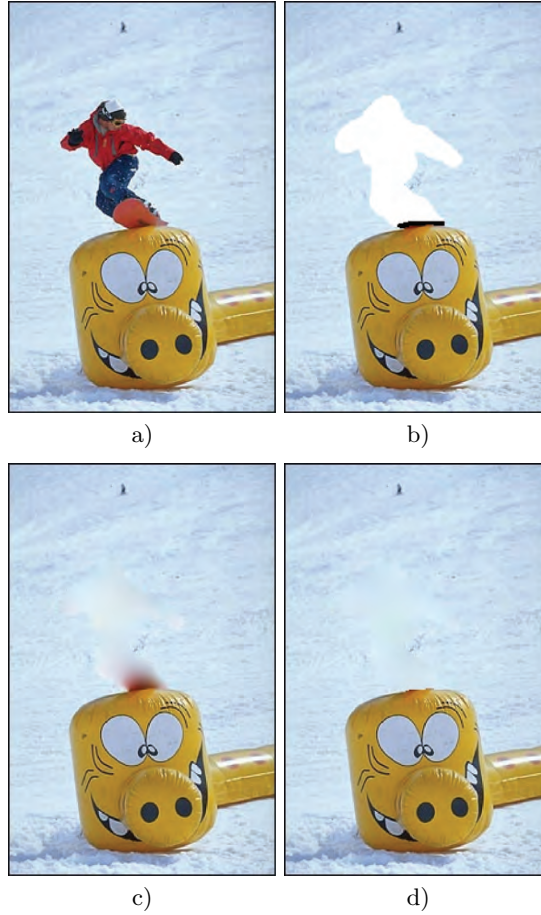


Fig. 12. a) A photo taken by Espen Lystad, a well-known snowboard photographer in Norway. b) The image with the inpainting region marked. The Neumann boundary is black. c) The restored image only using Dirichlet boundary condition. d) The restored image using Dirichlet and Neumann boundary conditions. (Color images in Figure A.8.)

4 Conclusion

In this work, we have proposed a method which uses two second order equations to do image inpainting. The equations used here are similar to the

equations used in [2] and [3]. By imposing the zero divergence condition which was not imposed in [2], it seems that our methods are able to produce better results when the inpainting region is rather large in diameter.

It is an interesting problem to study the existence and uniqueness for the solution for the equations we used. We have observed numerically that the gradient flow equations for (5) and (10) seem to have stable and unique solutions under the condition that the initial values are fixed.

References

1. A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Process.*, 13(9), 2004.
2. C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Trans. Image Process.*, (10):1200–1211, 2000.
3. M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-Stokes, fluid dynamics and image and video inpainting. In *Proc. Conf. Comp. Vision Pattern Rec.*, pages 355–362, 2001.
4. M. Bertalmio, G. Sapiro, C. Ballester, and V. Caselles. Image inpainting. *Computer Graphics, SIGGRAPH*, 2000.
5. M. Bertalmio, L. Vese, G. Sapiro, and O. Osher. Simultaneous texture and structure image inpainting. *IEEE Trans. Image Process.*, 10(8), 2003.
6. V. Caselles, S. Masnou, J.-M. Morel, and Catalina Sbert. Image interpolation. In *Séminaire sur les Équations aux Dérivées Partielles, 1997–1998*, pages Exp. No. XII, 15. École Polytech., Palaiseau, 1998.
7. T. F. Chan and J. Shen. Variational restoration of nonflat image features: Models and algorithms. *SIAM J. Appl. Math.*, 61(4):1338–1361, 2000.
8. T. F. Chan, S. H. Kang, and J. Shen. Euler’s elastica and curvature-based inpainting. *SIAM J. Appl. Math.*, 63(2):564–592, 2002.
9. T. F. Chan and J. Shen. Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.*, 62(3):1019–1043, 2002.
10. T. F. Chan, J. Shen, and L. Vese. Variational PDE models in image processing. *Notices Am. Math. Soc.*, 50(1):14–26, 2003.
11. F. Guichard and L. Rudin. Accurate estimation of discontinuous optical flow by minimizing divergence related functionals. *Proceedings of International Conference on Image Processing, Lausanne, September, 1996*, pages 497–500, 1996.
12. H. Grossauer and O. Scherzer. Using the complex Ginzburg-Landau equation for digital inpainting in 2D and 3D. In *Sacle space method in computer vision*, Lectures notes in Computer Sciences 2695. Springer, 2003.
13. J. Shen. Gamma-convergence approximation to piecewise constant mumford-shah segmentation. Tech. rep. (05-16), UCLA, Applied mathematics, 2005.
14. J. Weickert. *Anisotropic Diffusion in Image Processing*. Stuttgart, B. G. Teubner, 1998.
15. J. Lie, M. Lysaker, and X.-C. Tai. A binary level set model and some applications to image processing. *IEEE Trans. Image Process.*, to appear.
16. J. Lie, M. Lysaker, and X.-C. Tai. A variant of the levelset method and applications to image segmentation. *Math. Comp.*, to appear.

17. T. Lu, P. Neittaanmaki, and X.-C. Tai. A parallel splitting up method and its application to Navier-Stoke equations. *Appl. Math. Lett.*, 4:25–29, 1991.
18. T. Lu, P. Neittaanmaki, and X.-C. Tai. A parallel splitting up method for partial differential equations and its application to Navier-Stokes equations. *RAIRO Math. Model. and Numer. Anal.*, 26:673–708, 1992.
19. M. Lysaker, A. Lundervold, and X.-C. Tai. Noise Removal Using Fourth-Order Partial Differential Equation with Applications to Medical Magnetic Resonance Images in Space and Time. *IEEE Trans. Image Process.*, 12(12):1579–1590, 2003.
20. M. Lysaker, S. Osher, and X.-C. Tai. Noise removal using smoothed normals and surface fitting. *IEEE Trans. Image Processing*, 13(10):1345–1457, 2004.
21. S. Masnou. Disocclusion: a variational approach using level lines. *IEEE Trans. Image Process.*, 11(2):68–76, 2002.
22. J. Weickert, B. H. Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. Image Process.*, 7:398–409, 1998.

Error Analysis for H^1 Based Wavelet Interpolations

Tony F. Chan¹, Hao-Min Zhou², and Tie Zhou³

¹ Department of Mathematics, University of California, Los Angeles, CA 90095.
U.S.A. chan@math.ucla.edu

² School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332,
U.S.A. hmzhou@math.gatech.edu

³ School of Mathematical Sciences, Peking University, Beijing 100871, P.R. China.
tzhou@math.pku.edu.cn

Summary. We rigorously study the error bound for the H^1 wavelet interpolation problem, which aims to recover missing wavelet coefficients based on minimizing the H^1 norm in physical space. Our analysis shows that the interpolation error is bounded by the second order of the *local* sizes of the interpolation regions in the wavelet domain.

1 Introduction

In this paper, we investigate the theoretical error estimates for variational wavelet interpolation models.

The wavelet interpolation problem is to calculate unknown wavelet coefficients from given coefficients. It is similar to the standard function interpolations except the interpolation regions are defined in the wavelet domain. This is because many images are represented and stored by their wavelet coefficients due to the new image compression standard JPEG2000. The wavelet interpolation is one of the essential problems of image processing and closely related to many tasks such as image compression, restoration, zooming, inpainting, and error concealment, even though the term “interpolation” does not appear very often in those applications. For instance, wavelet inpainting and error concealment are to fill in (interpolate) damaged wavelet coefficients in given regions in the wavelet domain. Wavelet zooming is to predict (extrapolate) wavelet coefficients on a finer scale from a given coarser scale coefficients.

A major difference between wavelet interpolations and the standard function interpolations is that the applications of wavelet interpolations often impose regularity requirements of the interpolated images in the pixel domain, rather than the wavelet domain. For example, natural images (not including textures) are often viewed as piecewise smooth functions in the pixel domain.

This makes the wavelet interpolations more challenging as one usually cannot directly use wavelet coefficients to ensure the required regularity in the pixel domain. To overcome the difficulty, it seems natural that one can use optimization frameworks, such as variational principles, to combine the pixel domain regularity requirements together with the popular wavelet representations to accomplish wavelet interpolations.

A different reason for using variational based wavelet interpolations is from the recent success of partial differential equation (PDE) techniques in image processing, such as anisotropic diffusion for image denoising [25], total variation (TV) restoration [26], Mumford-Shah and related active contour segmentation [23, 10], PDE or TV image inpainting [1, 8, 7], and many more that we do not list here. Very often these PDE techniques are derived from variational principles to ensure the regularity requirements in the pixel domain, which also motive the study of variational wavelet interpolation problems.

Many variational or PDE based wavelet models have been proposed. For instance, Laplace equations, derived from H^1 semi-norm, has been used for wavelet error concealment [24], TV based models are used for compression [5, 12], noise removal [19], post-processing to remove Gibbs' oscillations [16], zooming [22], wavelet thresholding [11], wavelet inpainting [9], l^1 norm optimization for sparse signal recovery [3, 4], anisotropic wavelet filters for denoising [14], variational image decomposition [27]. These studies have demonstrated promising results, which show clear advantages of the combinations of wavelet and variational PDE strategies over the traditional methods.

Despite of the remarkable results obtained in combining variational PDE's with wavelets, the theoretical understandings for those models remain limited, specially for the nonlinear TV based models. Most of the existing studies are focused on the existence and uniqueness (or non-uniqueness) of the solutions of the variational wavelet models. A few recent investigations have been conducted to address the recover properties, including the well-known results reported in [3], in which a probabilistic theory for the exact recovery conditions of sparse signals based on random frequency samples has been developed. In [4], the authors have also studied the reconstruction error in probability sense for the random sampling model based on l^1 minimization of the Fourier frequencies for functions with certain power-law decaying frequencies.

To quantify the interpolation ability of those variational wavelet interpolation models, it is highly desirable to obtain rigorous error estimates, similar to the error bounds for the standard variational image inpainting problems as studied by Chan-Kang in [6] in which the recovery error is bounded by the square of local width of the inpainting region in the pixel domain if H^1 minimization is employed. However, this error analysis for variational wavelet interpolation models often faces different difficulties. For instance, the missing wavelet coefficients in wavelet space could have global influence in physical space, and the imposed regularity (smoothness) requirements are in physical space while the interpolations are performed in the wavelet space. Therefore, how to precisely estimate the regularity requirements in the wavelet space

becomes the key to carry out the analysis. This might be very challenging, specially for the nonlinear TV models in which one cannot characterize the TV semi-norm by size properties on wavelet coefficients [20]. For these reasons, such error estimates are still lacking for most of the variational wavelet interpolation models. This paper is our first attempt in gaining an understanding of those models from the error estimate perspective. We investigate the error bound for the H^1 wavelet interpolation model. Similar to the results in [6], our analysis shows that the error bound depends quadratically on the local size of the interpolation regions in wavelet domain. The ultimate goal of our current study is to develop a general strategy and theory to study error estimates for general variational PDE based wavelet models in image processing. We hope the results obtained in this paper can shed some lights for the general theory.

The rest of the paper is arranged as following: in the next section, we present the general variational wavelet interpolation models. The error estimate is given in Section 3.

2 Variational Wavelet Interpolation Models

In this section, we give the variational models of wavelet interpolations, which have been used in many applications. To better illustrate the analysis and simplify the discussion, we restrict ourselves to the one dimensional models. The results can be extended to higher dimensions with appropriate modifications.

We shall start with a brief review of continuous wavelet transforms to introduce notations that will be useful in this paper. Detailed wavelet theory can be found in many texts, such as [15, 28, 21, 17, 13]. A continuous wavelet transform is based on a selected real function $\psi(x) \in L^2(\mathbb{R})$, called wavelet function, satisfying,

$$C_\psi = 2\pi \int_0^{+\infty} \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < +\infty, \quad (1)$$

where $\hat{\psi}$ is the Fourier Transform of ψ . For requirements on how to select ψ , we refer to [15]. A family of wavelet functions is constructed by dilation and translations of $\psi(x)$ in the following format,

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right), \quad (2)$$

where $b \in \mathbb{R}$ is the translation variable and $a > 0$ the dilation variable. We denote as $a \in \mathbb{R}^+$, and $V = \mathbb{R}^+ \times \mathbb{R}$. In the wavelet literature, different dilation values of a often refer to the different resolutions or scales.

Let $z(x)$ be any function in $L^2(\mathbb{R})$, its continuous wavelet transform is defined by

$$\beta(a, b) = \int_{-\infty}^{+\infty} z(x) \psi_{a,b}(x) dx. \quad (3)$$

Similar to the Fourier transform, the wavelet transform is perfectly invertible, and the inverse wavelet transform is given by

$$z(x) = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} \frac{\beta(a, b)}{a^2} \psi_{a, b}(x) da db. \quad (4)$$

The continuous wavelet transform (3) provides a very redundant description of the function $z(x)$. For this reason, discrete wavelet transforms have been used more often in practice. To obtain the discrete wavelet transforms, one samples the continuous wavelet transform (3) at selected dyadic points. For example, a traditional (and also the most popular) selection takes $a_j = 2^j$ and $b_k = 2^j k$, where j, k are integers. This means that discrete wavelet coefficients are defined by

$$\beta_{j, k} = \beta(a_j, b_k) = \int_{-\infty}^{+\infty} z(x) \psi_{a_j, b_k}(x) dx, \quad (5)$$

and its reconstruction formula (discrete inverse wavelet transform) is given by

$$z(x) = \sum_{j, k} \beta_{j, k} \psi_{j, k}(x) = \sum_{j, k} \beta_{j, k} 2^{-\frac{j}{2}} \psi(2^{-j} x - k). \quad (6)$$

In the discrete wavelet representation (6), the wavelet functions $\psi_{a_j, b_k}(x)$ often form an orthonormal basis of L^2 space.

Wavelet transforms have been widely used in many applications, the most remarkable ones are in image processing such as compression, zooming, inpainting. A common challenge in those applications is that partial information of the discrete wavelet transforms $\beta(a_j, b_k)$ is not available for either deliberate (image compression) or involuntary (error concealment) reasons. For instance, the wavelet inpainting and error concealment consider problems that partial wavelet transforms are damaged or lost in the transmission or storage, and image compression algorithms record only selective, usually the significant, wavelet coefficients. Therefore, to restore the original images, one wants to recover the lost information based on the known coefficients. In image zooming or super-resolution, one wants to extend the information, which is only defined on a coarse grid, to a finer grid.

To solve these problems, one needs to interpolate the unavailable information from the known coefficients. To be mathematical precise, we describe the wavelet interpolation problem as following.

Let $z(x)$ be the original function having forward and inverse wavelet transforms defined by (5) and (6) respectively. If $I \subset V$ is a subset in which the discrete wavelet coefficients are not available, we denote

$$\alpha(a_j, b_k) = \begin{cases} \text{unknown} & \text{if } (a_j, b_k) \in I \\ \beta(a_j, b_k) & \text{if } (a_j, b_k) \in I^c \end{cases},$$

where I^c is the complement of I in V , as the wavelet transform for the to-be recovered function $u(x)$. The wavelet interpolation problem is to approximate the original function $z(x)$ by reconstructing $u(x)$ or $\alpha(a_j, b_k)$ on I from $\beta(a_j, b_k)$ on I^c .

Many different approaches have been proposed to achieve this goal. In this paper, we consider one strategy that uses variational principles in the optimization framework to help controlling the regularity of the interpolation. Let $F(\alpha)$ be an energy functional associated with $u(x)$. The variational wavelet interpolation problem is posed in the following form:

$$\min_{\alpha(a_j, b_k), (a_j, b_k) \in I} F(\alpha), \text{ subject to } \alpha(a_j, b_k) = \beta(a_j, b_k), \text{ for } (a_j, b_k) \in I^c \quad (7)$$

Different energy functionals $F(\alpha)$ have been proposed. For example, the l^1 norm of the coefficients $\|\alpha\|_1$ has been used to recover sparse signals [3, 4]. The H^1 semi-norm $\|\nabla_x u\|_2^2$ is used in the error concealment algorithm [24]. The popular TV semi-norm $\|\nabla_x u\|_1$ has been used by different groups to wavelet inpainting [9], thresholding [11], compression [12], zooming [18, 22], and restoration [2, 16, 19]. Many of these models have achieved remarkable success in their applications. However, theoretical understandings are still limited, especially for the models using H^1 or TV norms. Most of the existing analysis is related to the existence and non-uniqueness of the minimizers. And it does not provide quantitative understandings on why the models work well. In this paper, we investigate the error estimate for the missing information recovery and hope to explain the observations being made in these applications.

3 Recovery Bound for the H^1 Model

In this paper, we focus on the H^1 variational wavelet interpolation model, which uses

$$F(\alpha) = \int |\nabla_x u(x, \alpha)|^2 dx, \quad (8)$$

in the wavelet interpolation model (7).

To simplify the analysis, we assume that the functions $u(x)$ and $z(x)$ are defined on an infinite domain with compact supports, which can be achieved by extending to the outside of the given finite regions to zero values smoothly. Under this assumption, the boundary treatment becomes trivial and we omit it in this paper.

We shall start the analysis by decomposing of the interpolation subset I into simple connected regions for each resolution, which become simple subintervals in one dimension. Given the structure of the space V , one can easily write

$$V = \bigcup_{a \in \mathbb{R}^+} \{(c, d) \in V | c = a\} = \bigcup_{a \in \mathbb{R}^+} V_a.$$

Subspaces V_a correspond to different resolutions or scales for different dilation values of a in the wavelet space.

For a given resolution with fixed value of a , we define

$$I_a = I \bigcap V_a,$$

which is the restriction of I onto the subspace V_a . It is easy to see that I_a is the subset to be interpolated on the resolution a . This leads to

$$I = \bigcup_a I_a,$$

which simply states that the interpolation subset I can be decomposed into subsets I_a on different resolutions a . It is worth to remind that a is taken as discrete values $a_j = 2^j$ in the discrete wavelet interpolation problem. In the one dimensional case, it is obvious that I_{a_j} is just a measurable subset of $\mathbb{R}$. One can further divide it into disjoint subintervals

$$I_{a_j} = \bigcup_m I_{a_j,m} = \bigcup_m (b_{a_j,m}^1, b_{a_j,m}^2),$$

with

$$I_{a_j,m} \bigcap I_{a_j,n} = \phi, \quad \text{for } m \neq n,$$

and ϕ is the empty set. In other words, $I_{a_j,m} = (b_{a_j,m}^1, b_{a_j,m}^2)$ is a simple connected subregion to be interpolated on the resolution a_j . The wavelet coefficients at two ending points $\alpha(a_j, b_{a_j,m}^1), \alpha(a_j, b_{a_j,m}^2)$ are known to be $\beta(a_j, b_{a_j,m}^1)$ and $\beta(a_j, b_{a_j,m}^2)$ respectively. We call the width of the subinterval $|I_{a_j,m}| = |b_{a_j,m}^2 - b_{a_j,m}^1|$ the local size of the interpolation region. We denote

$$\epsilon = \inf_{a_j} \max_m |I_{a_j,m}|,$$

which is the largest width of all subinterval, or the maximum value of the local sizes of the interpolation regions.

Theorem 1. Assume $u(x)$ is a minimizer of (8). If the wavelet function $\psi(x)$ is in C^2 and $\frac{d^2\psi(x)}{dx^2} \in L^2$, then the continuous wavelet transform $\alpha(a, b)$ of $u(x)$ is C^2 with respect to b , and satisfies

$$\begin{cases} -\Delta_b \alpha(a_j, b_k) = 0, & \text{for all sample points } (a_j, b_k) \in I_{a_j,m} \\ \alpha(a_j, b_{a_j,m}^1) = \beta(a_j, b_{a_j,m}^1), & \alpha(a_j, b_{a_j,m}^2) = \beta(a_j, b_{a_j,m}^2), \end{cases} \quad (9)$$

and

$$|\Delta_b \alpha(a_j, b)| \leq a_j^{-1} \|z\|_{H^1} \|\psi\|_{H^1}, \quad (10)$$

where $\Delta_b = \frac{\partial^2}{\partial b^2}$ is the Laplace operator with respect to b for each fixed resolution a_j , and $\|\cdot\|_{H^1}$ is the standard H^1 semi norm.

Proof From the definition

$$\alpha(a, b) = \int u(x) \psi_{a,b}(x) dx,$$

we have

$$\Delta_b \alpha(a, b) = \int u(x) \Delta_b \psi_{a,b}(x) dx.$$

Using the dilation and translation structure (2) of $\psi_{a,b}(x)$, we observe

$$\nabla_x \psi_{a,b}(x) = -\nabla_b \psi_{a,b}(x), \quad \text{and} \quad \Delta_x \psi_{a,b}(x) = \Delta_b \psi_{a,b}(x).$$

These lead to

$$\Delta_b \alpha(a, b) = \int u(x) \Delta_x \psi_{a,b}(x) dx = \int \Delta_x u(x) \psi_{a,b}(x) dx < \infty, \quad (11)$$

which is continuous with respect to b .

Let us denote γ_{a_j, b_k} a unit vector taking the only nonzero value at a sample point (a_j, b_k) . We consider the partial directional derivative of $(\partial_\alpha F)(\gamma_{a_j, b_k})$ defined by

$$\begin{aligned} (\partial_\alpha F)(\gamma_{a_j, b_k}) &= \lim_{\epsilon \rightarrow 0} \frac{1}{\epsilon} (F(\alpha + \epsilon \gamma_{a_j, b_k}) - F(\alpha)) \\ &= \lim_{\epsilon \rightarrow 0} \frac{1}{\epsilon} \int (|\nabla_x(u(x, \alpha + \epsilon \gamma_{a_j, b_k}))|^2 - |\nabla_x u(x, \alpha)|^2) dx \\ &= \int 2 \nabla_x u(x) \nabla_x \psi_{a_j, b_k}(x) dx \\ &= -2 \int \Delta_x u(x) \psi_{a_j, b_k}(x) dx. \end{aligned}$$

It is known from calculus of variation that the minimizer of (8) must satisfy

$$(\partial_\alpha F)(\gamma_{a_j, b_k}) = 0,$$

which implies

$$\int \Delta_x u(x) \psi_{a_j, b_k}(x) dx = 0, \quad \text{for any sample point } (a_j, b_k) \in I_{a_j, m}.$$

This is the Euler-Lagrange equation for the variational problem in wavelet space. From this equation and (11), we get (9).

We also have

$$\begin{aligned}
|\Delta_b \alpha(a_j, b)| &= \left| \int u(x) \Delta_b \psi_{a_j, b}(x) dx \right| \\
&= \left| \int u(x) \Delta_x \psi_{a_j, b}(x) dx \right| \\
&= \left| - \int \nabla_x u(x) \nabla_x \psi_{a_j, b}(x) dx \right| \\
&\leq \left(\int |\nabla_x u(x)|^2 dx \right)^{\frac{1}{2}} \left(\int |\nabla_x \psi_{a_j, b}(x)|^2 dx \right)^{\frac{1}{2}} \\
&= a_j^{-1} \|u(x)\|_{H^1} \|\psi\|_{H^1}.
\end{aligned}$$

Since $u(x)$ is a minimizer, we must have $\|u(x)\|_{H^1} \leq \|z(x)\|_{H^1}$ which completes the estimate (10) and the proof the theorem.

Theorem 2. *If the wavelet function $\psi(x)$ is in C^2 and $\frac{d^2 \psi(x)}{dx^2} \in L^2$, then the discrete wavelet transform $\alpha(a_j, b_k)$ of the minimizer $u(x)$ of (8) satisfies*

$$|\alpha(a_j, b_k) - \beta(a_j, b_k)| \leq 2a_j^{-1} \epsilon^2 \|z\|_{H^1} \|\psi\|_{H^1}. \quad (12)$$

Proof For each fixed resolution a_j , we define

$$g(b) = \alpha(a_j, b) - \beta(a_j, b),$$

which is C^2 with respect to b . Let us consider this function on the interpolation interval $I_{a_j, m}$. The interpolation problem ensures that $g(b)$ vanishes at two ending points of $I_{a_j, m}$ because $\alpha(a_j, b)$ and $\beta(a_j, b)$ take the same values, i.e.

$$g(b_{a_j, m}^1) = g(b_{a_j, m}^2) = 0.$$

Given any one point $b \in I_{a_j, m}$, we have Taylor expansions,

$$g(b_{a_j, m}^1) = g(b) + g'(b)(b_{a_j, m}^1 - b) + \frac{1}{2} g''(\xi_1)(b_{a_j, m}^1 - b)^2,$$

and

$$g(b_{a_j, m}^2) = g(b) + g'(b)(b_{a_j, m}^2 - b) + \frac{1}{2} g''(\xi_2)(b_{a_j, m}^2 - b)^2,$$

where ξ_1 and ξ_2 are two points in $I_{a_j, m}$. Thus

$$\begin{aligned}
g(b) &= g(b) - \frac{b_{a_j, m}^2 - b}{b_{a_j, m}^2 - b_{a_j, m}^1} g(b_{a_j, m}^1) - \frac{b - b_{a_j, m}^1}{b_{a_j, m}^2 - b_{a_j, m}^1} g(b_{a_j, m}^2) \\
&= \frac{1}{2} (g''(\xi_1)(b_{a_j, m}^1 - b)^2 + g''(\xi_2)(b_{a_j, m}^2 - b)^2) \\
&\leq \max_{\xi \in I_{a_j, m}} |g''(\xi)| \epsilon^2
\end{aligned}$$

Similar to the proof of (10), we obtain

$$\begin{aligned}
|g''(b)| &= |\Delta_b(\alpha(a_j, b) - \beta(a_j, b))| \\
&= \left| \int (u(x) - z(x)) \Delta_b \psi_{a_j, b}(x) dx \right| \\
&= \left| \int (u(x) - z(x)) \Delta_x \psi_{a_j, b}(x) dx \right| \\
&= \left| - \int \nabla_x (u(x) - z(x)) \nabla_x \psi_{a_j, b}(x) dx \right| \\
&\leq 2a_j^{-1} \|z(x)\|_{H^1} \|\psi(x)\|_{H^1},
\end{aligned}$$

which completes the proof.

We remark that for multi-dimensional wavelet interpolation problems, Theorems 1 and 2 still hold with the understanding that $I_{a, m}$ becomes multi-dimensional regions. We will not address them in detail in this paper.

4 A Numerical Example

The estimate obtained in Section 3 shows that the approximation error for the H^1 wavelet interpolation model is bounded quadratically by the local size of the interpolation regions. In this section, we compute the H^1 wavelet interpolations of a simple function

$$z(x) = \sin(4\pi x) \quad x \in (0, 1).$$

To illustrate the quadratic rate, we arbitrarily select l consecutive low frequency coefficients to be interpolated. We note that the doubled number l corresponds to the doubled size ϵ of the local interpolation region. We measure the maximum approximation error in the coefficients defined by

$$\text{EIC} = \max_k |\alpha_{a_j, b_k} - \beta_{a_j, b_k}|.$$

And the error rate is calculated by

$$\text{rate} = \log_2 \left(\frac{\text{EIC}(2l)}{\text{EIC}(l)} \right).$$

The error and its rate for different number l are shown in Table 1. It clearly demonstrates that the error rate is close to 2 if the interpolation region is in the low frequencies. We also remark that our numerical experiments show that if the interpolation regions do not contain low frequencies, the error is much smaller than the quadratic estimate, which suggests that the rate may be improved if no low frequency coefficient is interpolated.

Conclusion and future work: The analysis shows that the recovery property of H^1 wavelet interpolation model is bounded quadratically by the local, not global, sizes of the interpolation regions in the wavelet domain,

l	EIC	rate
2	0.00045	1.76
4	0.00151	1.84
8	0.00541	1.89
16	0.02014	1.91
32	0.07565	1.71
64	0.24725	

Table 1. The maximum error in the coefficients for the H^1 wavelet interpolation model. The error rate indicate that error is bounded quadratically by the consecutive number of coefficients to be interpolated.

which is similar to the results for the pixel domain image inpainting problems reported in [6]. It explains that good restorations can be achieved if the local interpolation regions are small even if their total size is large. For instance, if the interpolation regions are randomly distributed as small disjoint regions in the wavelet domain, good interpolation computations are achieved even the total size of the interpolation regions is significant. On the contrary, if there is one large region to be interpolated, the error would be large in this region. This error bound is also consistent with many computations such as these reported in [24] and [9].

The results reported here are for H^1 based wavelet interpolation model. However it is well known that H^1 based model often over smooths edges in images. TV or other nonlinear energy based models can preserve the discontinuities better. The recovery bounds for those models are beyond the scope of this paper and we will not address them here.

Acknowledgement. The first author is supported in part by grants ONR N00014-06-1-0345, NIH U54 RR021813, and NSF CCF-0528586, the second author is supported in part by grant NSF DMS-0410062, and third author's research is supported by the National Basic Research Program of China under Grant 2003CB716101, National Science Foundation of China under Grant 60532080, 60372024, Key Project of Chinese Ministry of Education under Grant 306017, Engineering Research Institute of Peking University.

References

1. M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, *Image Inpainting*, Tech. Report, ECE-University of Minnesota, 1999.
2. E. Candès, and F. Guo. *Edge-preserving Image Reconstruction from Noisy Radon Data*, (Invited *Special Issue of the Journal of Signal Processing on Image and Video Coding Beyond Standards.*), 2001.
3. E. Candès, J. Romberg and T. Tao, *Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information*, Preprint: arXiv:math.GM/0409186, Sept. 2004.

4. E. Candès, and T. Tao, *Near Optimal Signal Recovery From Random Projections and Universal Encoding Strategies*, Preprint, submitted to IEEE Information Theory, Oct. 2004.
5. A. Chambolle, R. A. DeVore, N.-Y. Lee, and B. J. Lucier. *Nonlinear wavelet image processing: variational problems, compression and noise removal through wavelet shrinkage*. IEEE Trans. Image Processing, 7(3):319–335, 1998.
6. T. F. Chan and S. H. Kang, *Error Analysis for Image Inpainting*, to appear Journal of Mathematical Imaging and Vision, 2006.
7. T. F. Chan, S. H. Kang and J. Shen, *Euler's Elastica and Curvature Based inpainting*, SIAM J. Appl. Math., 63(2) (2002), 564-592.
8. T. F. Chan and J. Shen, *Mathematical Models for Local Non-Texture Inpainting*, SIAM J. Appl. Math., 62(3) (2002), 1019-1043.
9. T. F. Chan, J. Shen, and H. M. Zhou, *Total Variation Wavelet Inpainting*, to appear in J. of Math. Imaging and Vision.
10. T. F. Chan and L. Vese, *Active Contour Without Edges* IEEE Tran. on Image Proc., 10(2), Feb. 2001, pp 266-277.
11. T. F. Chan and H. M. Zhou, *Total Variation Wavelet Thresholding*, submitted to J. Comp. Phys..
12. T. F. Chan and H. M. Zhou, *Optimal Constructions of Wavelet Coefficients Using Total Variation Regularization in Image Compression*, CAM Report, No. 00-27, Dept. of Math., UCLA, July 2000.
13. C. K. Chui, *Wavelet: A Mathematical Tool for Signal Analysis*, SIAM, 1997.
14. C. K. Chui and J. Wang, *Wavelet-based Minimal-Energy Approach to Image Restoration*, submitted to ACHA.
15. I. Daubechies. *Ten lectures on wavelets*. SIAM, Philadelphia, 1992.
16. S. Durand and J. Froment, *Artifact Free Signal Denoising with Wavelets*, in Proceedings of ICASSP'01, volume 6, 2001, pp. 3685-3688.
17. E. Hernandez and G. Weiss, *A First Course on Wavelets*, CRC Press, 1996.
18. F. Malgouyres, *Increase in the Resolution of Digital Images: Variational Theory and Applications*, Ph.D. thesis, Ecole Normale Supérieure de Cachan, 2000, Cachan, France.
19. F. Malgouyres, *Mathematical Analysis of a Model Which Combines Total Variation and Wavelet for Image Restoration*, Journal of information processes, 2:1, 2002, pp 1-10.
20. Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, vol 22 of *University Lecture Seires*, AMS, Providence, 2001.
21. S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1998.
22. L. Moisan, *Extrapolation de Spectre et Variation Totale Ponderee*, actes du GRETSI, 2001.
23. D. Mumford and J. Shah, *Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems*, Comm, Pure Appl. Math. 42, 1989, pp577-685.
24. Yan Niu and T. Poston, *Harmonic postprocessing to conceal for transmission errors in DWT coded images*. preprint, Institute of Eng. Sci., National Univ. of Singapore, 2003.
25. P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE T PATTERN ANAL. 12: (7), July, 1990, pp629-639.
26. L. Rudin, S. Osher and E. Fatemi, *Nonlinear Total Variation Based Noise Removal Algorithms*, Physica D, Vol 60(1992), pp. 259-268.

34 T.F. Chan, H.-M. Zhou, T. Zhou

- 27. J.L. Starck, M. Elad and D. Donoho. *Image Decomposition via the Combination of Sparse Representations and a Variational Approach*. to appear in the IEEE Trans. Image Processing.
- 28. G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesley, MA, 1996.

Image Dejittering Based on Slicing Moments

Sung Ha Kang¹ and Jianhong (Jackie) Shen²

¹ Department of Mathematics, University of Kentucky, Lexington, KY 40515, USA. E-mail: skang@ms.uky.edu

² School of Mathematics, University of Minnesota, Minneapolis, MN 55455. E-mail: jhshen@math.umn.edu

Summary. Since the celebrated works of Rudin, Osher, and Fatemi (*Physica D*, **60**:259-268, 1992), the space of functions with bounded variations (BV) has become a powerful mathematical model for approximating generic image signals in contemporary imaging and vision sciences. Motivated by the important application of image and video dejittering, we study the mathematical properties of the slicing moments of BV images. The regularity characterization leads to a novel model for the dejittering problem based upon the Bayesian/Tikhonov principle. Analytical as well as computational results are developed for this novel dejittering approach.

Key words: Bounded variation, slicing moments, Bayesian, inverse problem, dejittering, variational, regularization, existence.

1 Introduction

The growing popularity of image processing and vision analysis within the mathematics community has been determined by two basic facts:

- (a) images and visual signals are first of all *functions* [7], and
- (b) understanding the *patterns* [19] of these functions is fundamentally a mathematical problem. Both visual perception (e.g., for robots) and image analysis (e.g., medical CT or MRI images) are about understanding various *patterns* [19] which are often refer to as geometrical, algebraic, topological, or stochastic invariants.

Thus, processing images or visual signals amounts to analyzing a special class of functions called *images*, which serves as the mathematical foundation of image processing.

In recent two decades, the marriage of image processing, vision analysis, and mathematics has nurtured numerous exciting discoveries as well as revived various classical subjects, e.g., wavelets, multiresolution analysis, oscillatory patterns, fractals, moving fronts, multiphase problems with free boundaries,

and Gibbs' random fields, just to name a few [10, 12, 17, 20, 21, 28]. Mathematics has provided the solid ground for solving many challenging imaging and vision problems in a unified and mass-production manner. At the same time, countless emerging applications of imaging and vision technologies in this information era have provided fertile soils for nurturing new problems and theories in mathematics. The recent expository article [9] and research monograph [7] provide further details along this line.

This current work is easily embedded into this general picture of contemporary mathematical image and vision analysis (Miva). Inspired by an important application called image (or video) dejittering, we introduce and explore the properties of the slicing moments of multi-dimensional functions with *bounded variations* (BV).

The BV image model was first introduced into image analysis in the celebrated work of Rudin, Osher, and Fatemi [21]. It has become one of the most powerful image models that reach a good balance between geometric fidelity and computational complexity (e.g., [1, 3, 5, 6, 8, 22, 24, 27]). Numerous applications have shown that except for oscillatory textures of small amplitudes [2, 25], the BV image model performs sufficiently well in characterizing visually important geometric features like edges.

Motivated by the image dejittering problem, in the current paper, we first introduce and study the properties of the slicing moments of BV functions, and then propose a novel dejittering model based upon the idea of moment regularization. Our mathematical framework is intentionally kept general (in terms of dimensions and assumptions), and aims at contributing to solving many other problems in related applied sciences.

As shown in Fig. 1, image jittering occurs when the slices of a high dimensional image signal are *randomly* displaced along the slicing hyperplane (e.g., the horizontal line in 2D). Three major technological areas where jittering frequently arises are: (a) *video jittering* due to the corruption of synchronization signals in analog video tapes; (b) *video interlacing* due to the temporal difference between the fast motions of objects in a scene and the refreshing speed of a digital display device; and (c) *slice jittering* in imaging devices such as CT (computer tomography) and MRI (magnetic resonance imaging) scanning, when patients or devices undergo random spatial displacements during an imaging acquisition process.

To restore an ideal image u from its jittered version u_J is the problem of *image dejittering*. For corrupted analog videos, in [15, 16], Kokaram and his colleagues first explored dejittering methods that only rely upon the jittered video images instead of other irrelevant tape information. Such approaches are said to be *intrinsic* in contrast with most conventional video dejittering techniques, which employ extra image-irrelevant information. In [23], the second author developed an intrinsic variational dejittering model based on Bayesian estimation theory. In [14], the two authors further proposed a flexible two-step model called “bake and shake” for intrinsic image dejittering using nonlinear diffusion partial differential equations.



Fig. 1. (a) Ideal image $u(x, y)$. (b) Randomly jittered image $u_J(x, y)$. (Color images in Figure A.9.)

The aforementioned works could be considered as “differential” since they all depend upon the characterizations of local image structures. The current work, therefore, distinguishes in its “integral” nature since slicing moments are integrated quantities. In general, integral methods are more robust to small perturbations. Furthermore, integrated quantities like moments naturally achieve dimensionality reduction and gain substantial computational efficiency.

The paper is organized as follows. In Section 2, we first introduce the notion of slicing moments for high-dimensional BV images, and prove that they generally inherit the BV regularity. In Section 3, based on the regularity of slicing moments as well as Bayesian estimation theory, we propose a novel variational dejittering model in arbitrary dimensions, and establish its well-posedness by showing the existence and uniqueness of the optimal solution. In Section 4, algorithm and numerical examples are presented to demonstrate the performance of the new dejittering model. A brief conclusion is made in Section 5.

2 Slicing Moments of BV Functions

In this section, we first show that the slicing moments of a typical BV image is also a BV function, which enables us to employ the Bayesian restoration framework for image dejittering [7]. In this paper, we shall study BV functions in $\mathbb{R}^n$ which are compactly supported and nonnegative:

$$BV_c^+ = BV_c^+(\mathbb{R}^n) = \{v \in L^1(\mathbb{R}^n) \mid v \geq 0, \text{ compactly supported, and } \int_{\mathbb{R}^n} |Dv| < \infty\}.$$

Nonnegativity is a plausible assumption in imaging and vision since physically image values represent photon counts. Recall that the total variation (TV) Radon measure is defined by, for any open domain $U \subseteq \mathbb{R}^n$,

$$\int_U |Dv| = \sup_{\mathbf{g} \in C_c^1(U, \mathcal{B}^n)} \int_U v \operatorname{div}(\mathbf{g}) dz, \quad \text{with } dz = dz_1 \cdots dz_n, \quad (1)$$

where $\mathcal{B}^n$ denotes the n -dimensional unit ball centered at the origin in $\mathbb{R}^n$. Fixing any $d = 0, 1, \dots, n-1$, we write $z = (x, y) \in \mathbb{R}^n$ with

$$x = (z_1, \dots, z_{n-d}) \in \mathbb{R}^{n-d} \quad \text{and} \quad y = (z_{n-d+1}, \dots, z_n) \in \mathbb{R}^d.$$

For any multi-exponent $\alpha = (\alpha_1, \dots, \alpha_{n-d}) \in \{0, 1, 2, \dots\}^{n-d}$, define x^α to be

$$x^\alpha = z_1^{\alpha_1} z_2^{\alpha_2} \cdots z_{n-d}^{\alpha_{n-d}} \in \mathbb{R}.$$

Definition 1 (Slicing Moments). *Given an image $u \in BV_c^+$ and an exponent α , the slicing moment of u of codimension d is defined by*

$$m_d(y|u, \alpha) = \int_{\mathbb{R}^{n-d}} x^\alpha u(x, y) dx. \quad (2)$$

Notice that m_d is a function in $\mathbb{R}^d$ for any given u and α . The integral is indeed well defined since $u \in BV_c^+$ is assumed to be compactly supported. Fig. 2 shows an example of slicing moments of a simple image with dimension $n = 2$ and codimension $d = 1$. The image on the left panel is a synthetic

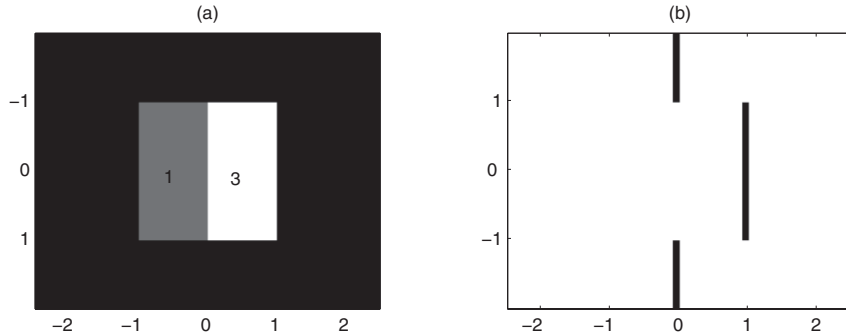


Fig. 2. (a) Image sample u . (b) The (horizontal) slicing moment $m_1(y | u, 1)$.

BV image, and the graph plotted on the right panel is its (horizontal) slicing moment with $\alpha = 1$ and $d = 1$. It is clear that the slicing moment is piecewise constant and still a BV function of y . If image (a) is jittered, the moment function in (b) would become noisy, and effective noise estimation can reveal

the important information about the unknown jitters. This is the key observation leading to our novel dejittering model later. We now first show that the slicing moment function is also a BV function provided that the given image u is. This theorem is crucial for our new model, since it allows regularization techniques for *degraded* BV functions [21].

Theorem 1. *For any given image $u \in BV_c^+(\mathbb{R}^n)$, codimension $d \in \{0, 1, \dots, n-1\}$, and multi-exponent $\alpha \in \{0, 1, \dots\}^{n-d}$,*

$$m_d(y|u, \alpha) \in BV_c(\mathbb{R}^d).$$

Proof. We show that m_d is compactly supported, belongs to $L^1(\mathbb{R}^d)$, and $\int |Dm_d| < \infty$.

[1] Since u is compactly supported, there exists some $\gamma > 0$ such that

$$\text{supp } u \subseteq \{z \in \mathbb{R}^n : |z|_\infty = \max_{1 \leq i \leq n} |z_i| \leq \gamma\}. \quad (3)$$

In particular, for any $z = (x, y)$ with $x \in \mathbb{R}^{n-d}$ and $|y|_\infty > \gamma$, one has $u(x, y) = 0$ and

$$m_d(y|u, \alpha) = \int_{\mathbb{R}^{n-d}} x^\alpha u(x, y) dx = 0.$$

Therefore, $m_d(y|u, \alpha)$ is also compactly supported and

$$\text{supp } m_d(y|u, \alpha) \subseteq \{y \in \mathbb{R}^d : |y|_\infty \leq \gamma\}.$$

[2] Next, we show that $m_d \in L^1(\mathbb{R}^d)$. With $z = (x, y)$, one has

$$\begin{aligned} \int_{\mathbb{R}^d} |m_d(y|u, \alpha)| dy &= \int_{\mathbb{R}^d} \left| \int_{\mathbb{R}^{n-d}} x^\alpha u(x, y) dx \right| dy \\ &\leq \int_{\mathbb{R}^n} |x^\alpha| u(z) dz \quad (\text{by Fubini's Theorem}) \\ &= \int_{\{z: |x|_\infty \leq \gamma\}} |x^\alpha| u(z) dz \quad (\text{by (3)}) \\ &\leq \gamma^{|\alpha|} \int_{\mathbb{R}^n} u(z) dz < \infty, \quad (\text{since } u \in L^1(\mathbb{R}^n)) \end{aligned}$$

where $|\alpha| = \alpha_1 + \alpha_2 + \dots + \alpha_{n-d}$. Therefore, $m_d(y|u, \alpha) \in L^1(\mathbb{R}^d)$.

[3] By the definition of the TV Radon measure (1),

$$\int_{\mathbb{R}^d} |D m_d(y|u, \alpha)| = \sup_{\varphi \in C_c^1(\mathbb{R}^d, \mathcal{B}^d)} \int_{\mathbb{R}^d} m_d(y|u, \alpha) \text{div}_y(\varphi) dy, \quad (4)$$

where $y = (y_1, y_2, \dots, y_d)$, $\varphi = \varphi(y) = (\varphi_1, \dots, \varphi_d)$, and

$$\operatorname{div}_y(\varphi) = \partial_{y_1}\varphi_1 + \cdots + \partial_{y_d}\varphi_d.$$

For any fixed γ in (3), choose $\rho_\gamma(x) \in C_c^1(\mathbb{R}^{n-d})$ such that $\rho_\gamma(x) \in [0, 1]$ and

$$\rho_\gamma(x) = \begin{cases} 1, & \text{for } |x|_\infty \leq \gamma \\ 0, & \text{for } |x|_\infty > \gamma + 1 \end{cases}. \quad (5)$$

Then, $\forall z \in \mathbb{R}^n$ (with $x \in \mathbb{R}^{n-d}$ and $y \in \mathbb{R}^d$), one has

$$u(z) = u(x, y) \equiv u(x, y)\rho_\gamma(x). \quad (6)$$

For any given α and $\varphi(y) \in C_c^1(\mathbb{R}^d, \mathcal{B}^d)$, define a new flow on the entire space $\mathbb{R}^n$ by

$$\mathbf{g}(z) = \mathbf{g}(x, y) = (0^{n-d}, x^\alpha \varphi(y) \rho_\gamma(x)), \quad (7)$$

where 0^{n-d} denotes the origin of $\mathbb{R}^{n-d}$. Then,

$$\operatorname{div}(\mathbf{g}(z)) = \operatorname{div}_y(x^\alpha \rho_\gamma(x) \varphi(y)) = x^\alpha \rho_\gamma(x) \operatorname{div}_y(\varphi(y)). \quad (8)$$

Furthermore, by the definitions in (5) and (7),

$$\operatorname{supp} \mathbf{g} \subseteq \{x : |x|_\infty \leq \gamma + 1\} \times \operatorname{supp} \varphi(y),$$

implying that $\mathbf{g} \in C_c^1(\mathbb{R}^n, \mathbb{R}^n)$.

With $z = (x, y)$ and (5),

$$|\mathbf{g}(z)|_2 = |x^\alpha \rho_\gamma(x)| \cdot |\varphi(y)|_2 \leq (\gamma + 1)^{|\alpha|} \|\varphi\|_\infty, \quad (9)$$

where $\|\varphi\|_\infty = \sup_y |\varphi(y)|_2$. Therefore, $\gamma^{-|\alpha|} \mathbf{g} \in C_c^1(\mathbb{R}^n, \mathcal{B}^n)$.

For any test flow $\varphi(y) \in C_c^1(\mathbb{R}^d, \mathcal{B}^d)$, by Fubini's Theorem,

$$\begin{aligned} \int_{\mathbb{R}^d} m_d(y|u, \alpha) \operatorname{div}_y \varphi(y) dy &= \int_{\mathbb{R}^d} \left(\int_{\mathbb{R}^{n-d}} x^\alpha u(x, y) dx \right) \operatorname{div}_y \varphi(y) dy \\ &= \int_{\mathbb{R}^n} u(x, y) x^\alpha \operatorname{div}_y \varphi(y) dz \\ &= \int_{\mathbb{R}^n} u(x, y) \rho_\gamma(x) x^\alpha \operatorname{div}_y \varphi(y) dz \quad (\text{by (6)}) \\ &= \int_{\mathbb{R}^n} u(z) \operatorname{div} \mathbf{g}(z) dz \quad (\text{by (8)}) \\ &\leq \gamma^{|\alpha|} \int_{\mathbb{R}^n} |Du|. \quad (\text{by (9)}) \end{aligned}$$

Since φ is arbitrary and $u \in BV_c^+(\mathbb{R}^n)$, we conclude that

$$\int_{\mathbb{R}^d} |D m_d(y|u, \alpha)| \leq \gamma^{|\alpha|} \int_{\mathbb{R}^n} |Du| < \infty. \quad (10)$$

The proof is complete. $\square$

In particular when $\alpha = 0^{n-d}$, we have the following corollary for marginal projections, which is needed for later developments. (The term “marginal” has been motivated by the term “marginal distribution” in multivariate probability theory.)

Corollary 1 (Marginal Projections). *Define $M_d(y|u) = m_d(y|u, 0^{n-d})$ to be the marginal projection of codimension d . Then, $M_d(y|u) \in BV_c^+(\mathbb{R}^d)$, and*

$$\int_{\mathbb{R}^d} |D M_d(y|u)| \leq \int_{\mathbb{R}^n} |Du|. \quad (11)$$

Proof. Notice that $M_d \geq 0$ due to $u \geq 0$. Then, (11) follows from (10) for $\alpha = 0^{n-d}$. $\square$

In Theorem 1, the slicing moment functions have been shown to belong to the BV space. We now remark via the example in Fig. 2 that the BV regularity cannot be upgraded to the Sobolev regularity $W^{1,1}$. The image on the left panel of Fig. 2 is defined by, with $z = (x, y)$,

$$u(z) = \begin{cases} 0, & |z|_\infty > 1 \\ 1, & |z|_\infty \leq 1, \ x \leq 0 \\ 3, & |z|_\infty \leq 1, \ x > 0 \end{cases}.$$

For $\alpha = 1$, define the (horizontal) linear slicing moment $m(y|u) = m_1(y|u, 1) = \int_{\mathbb{R}} xu(x, y)dx$. Then, for $\forall y$ with $|y| > 1$, one has $m(y|u) \equiv 0$, and for $\forall y \in (-1, 1)$,

$$m(y|u) = \int_{-1}^0 xdx + \int_0^1 3xdx = \int_0^1 2xdx \equiv 1.$$

Therefore, as illustrated on the right panel of Fig. 2, $m(y|u) = 1_{|y| \leq 1}(y)$, and the signed total variation Radon measure is only expressible via Dirac’s delta function:

$$Dm(y|u) = \delta(y + 1) - \delta(y - 1),$$

which does not belong to $L^1(\mathbb{R})$. Thus, $m(y|u) \in BV(\mathbb{R}) \setminus W^{1,1}(\mathbb{R})$, and the regularity result in Theorem 1 is optimal.

For the application to *intrinsic* image dejittering, such regularity information will be the key to our novel model which is to be discussed next.

3 Moments Regularization for Image Dejittering

In this section, we apply the above regularity results to the problem of image dejittering.

3.1 Formulation of the jittering problem

In the language of inverse problems, dejittering is to invert the *forward problem* of jittering. Thus, we first propose a generic forward model for the jittering process.

Definition 2 (Jitter s). A q -dimensional jitter (field) on $\mathbb{R}^d$ is a random map:

$$s : \mathbb{R}^d \rightarrow \mathbb{R}^q, \quad y \rightarrow s(y),$$

such that, for any finite set of points $E \subseteq \mathbb{R}^d$,

$$\{s(y) \mid y \in E\},$$

are independent and identically distributed (i.i.d) random variables.

As an example, for any fixed $y \in \mathbb{R}^d$, jitter $s(y)$ could be subject to the Gaussian normal distribution $\mathcal{N}(0^q, \Sigma)$ with a covariance matrix Σ . In term of the probability density function (p.d.f.), one has

$$p(s = \hat{s}) = \frac{1}{\sqrt{(2\pi)^q |\Sigma|}} e^{-\frac{1}{2} \hat{s}^T \Sigma^{-1} \hat{s}}. \quad (12)$$

Definition 3 (Jittered Image u_J). Let $u \in BV_c^+(\mathbb{R}^n)$ and $d \in \{0, 1, \dots, n-1\}$. For any given $(n-d)$ -dimensional jitter $s(y)$ on $\mathbb{R}^d$, the jittered image u_J is defined to be :

$$u_J(z) = u_J(x, y) = u(x - s(y), y), \quad z \in \mathbb{R}^n, \quad x \in \mathbb{R}^{n-d}, \quad \text{and } y \in \mathbb{R}^d. \quad (13)$$

Definition 4 (Dejittering). The dejittering problem is the inverse problem of restoring the original image $u(z)$ from its jittered observation $u_J(z)$ (see Fig. 1).

3.2 Linear slicing moments and Bayesian inference

Definition 5 (Linear Slicing Moments). Let the codimension d linear moments $\mathbf{m}_d(y|u)$ for $u \in BV_c^+(\mathbb{R}^n)$ be the vectorial function

$$\mathbf{m}_d(y|u) = (m_d(y|u, e_1), \dots, m_d(y|u, e_{n-d})), \quad (14)$$

where $e_i = (0, \dots, 0, 1_{i^{\text{th}}}, 0, \dots, 0)$, $i = 1, \dots, n-d$. Equivalently, it is given by

$$\mathbf{m}_d(y|u) = \int_{\mathbb{R}^{n-d}} x u(x, y) dx, \quad x = (z_1, \dots, z_{n-d}).$$

By Theorem 1, one immediately has the following regularity.

Corollary 2. The linear slicing moment $\mathbf{m}_d(y|u)$ belongs to $BV_c(\mathbb{R}^d, \mathbb{R}^{n-d})$.

Notice that in terms of linear structures, one has $BV_c(\mathbb{R}^d, \mathbb{R}^{n-d}) = BV_c(\mathbb{R}^d, \mathbb{R})^{n-d} = BV_c(\mathbb{R}^d)^{n-d}$. As for the TV Radon measure in $BV_c(\mathbb{R}^d, \mathbb{R}^{n-d})$, we follow the general definition of total variations for product measures [11]. Recall that for any given p measures $\mu_1, \dots, \mu_p$ on a measurable space (X, Σ) (where Σ is a σ -algebra on X), the total variation $|\boldsymbol{\mu}|$ of the vectorial measure $\boldsymbol{\mu} = (\mu_1, \dots, \mu_p)$ is defined by ;

$$\text{For any } E \in \Sigma, \quad |\boldsymbol{\mu}|(E) = \sup_{\|\boldsymbol{\varphi}\|_\infty \leq 1} \sum_{i=1}^p \int_E \varphi_i d\mu_i = \sup_{\|\boldsymbol{\varphi}\|_\infty \leq 1} \int_E \boldsymbol{\varphi} \cdot d\boldsymbol{\mu},$$

where $\boldsymbol{\varphi}$ is a Σ -measurable vectorial function, and

$$\|\boldsymbol{\varphi}\|_\infty = \sup_{x \in X} |\boldsymbol{\varphi}|_2(x) = \sup_{x \in X} \sqrt{\varphi_1^2(x) + \dots + \varphi_p^2(x)}.$$

One symbolically writes $|\boldsymbol{\mu}| = \sqrt{\mu_1^2 + \dots + \mu_p^2}$. If there exists a (positive) measure v on (X, Σ) , such that all the Radon-Nikodym derivatives exist:

$$\rho_i = \frac{d\mu_i}{dv}, \quad i = 1, \dots, p,$$

then, $|\boldsymbol{\mu}|$ must be differentiable with respect to v , and

$$\frac{d|\boldsymbol{\mu}|}{dv} = |\boldsymbol{\rho}|_2 = \sqrt{\rho_1^2 + \dots + \rho_p^2}$$

or equivalently $|\boldsymbol{\mu}|(E) = \int_E |\boldsymbol{\rho}|_2 dv$ for any $E \in \Sigma$.

By this general framework, the natural total variation measure in the space $BV_c(\mathbb{R}^d, \mathbb{R}^{n-d})$ for $\mathbf{m}_d(y|u)$ is: for any Borel set $U \subseteq \mathbb{R}^d$,

$$\int_U |D \mathbf{m}_d(y|u)| = \int_U \left[\sum_{i=1}^{n-d} [Dm_d(y|u, e_i)]^2 \right]^{1/2}, \quad (15)$$

where e_i 's are as in (14). In particular, if $\mathbf{m}_d \in W^{1,1}(\mathbb{R}^d, \mathbb{R}^{n-d})$, one has

$$\int_U |D \mathbf{m}_d(y|u)| = \int_U \left[\sum_{i=1}^{n-d} [\nabla m_d(y|u, e_i)]^2 \right]^{1/2} dy. \quad (16)$$

By Corollary 2, $\mathbf{m}_d$ is a BV vectorial function under the definition in (15). In the following proposition, we consider the link between the linear slicing moment and the jitter $s(y)$.

Proposition 1. *Let $u_J(z)$ denote the jittered image generated from $u(z)$ by jitter $s(y)$ as in (13). Then, the linear slicing moment of u_J and u are connected by:*

$$\mathbf{m}_d(y|u_J) = \mathbf{m}_d(y|u) + s(y)M_d(y|u), \quad (17)$$

where $M_d(y|u)$ is the codimension d marginal projection of u as defined in Corollary 1.

Proof. It suffices to carry out the following computation,

$$\begin{aligned}
\mathbf{m}_d(y|u_J) &= \int_{\mathbb{R}^{n-d}} xu_J(x, y)dx = \int_{\mathbb{R}^{n-d}} xu(x - s(y), y)dx \\
&= \int_{\mathbb{R}^{n-d}} (t + s(y))u(t, y)dt \\
&= \int_{\mathbb{R}^{n-d}} tu(t, y)dt + s(y) \int_{\mathbb{R}^{n-d}} u(x, y)dx \\
&= \mathbf{m}_d(y|u) + s(y)M_d(y|u). \quad \square
\end{aligned}$$

Therefore, if the true image u were known, one could easily identify the jitter $s(y)$ by Proposition 1. In reality, only u_J and $\mathbf{m}_d(y | u_J)$ are directly available while u and $\mathbf{m}_d(y | u)$ are unknown. The following proposition shows that $M_d(y | u)$ is in fact directly readable from the jittered image u_J .

Proposition 2. *The marginal projection is jittering invariant, i.e.,*

$$M_d(y|u_J) = M_d(y|u).$$

The proof is straightforward since the Lebesgue measure dx is translation-invariant. Eqn. (17) now becomes

$$\mathbf{m}_d(y|u_J) = \mathbf{m}_d(y|u) + s(y)M_d(y|u_J). \quad (18)$$

To summarize, in terms of estimating the unknown linear slicing moment $\mathbf{m}_d(y | u)$, (which is equivalent to the estimation of the jitter $s(y)$), we have established the following two key ingredients in the framework of Bayesian inference [12, 18].

1. The prior model: Eqn. (15) specifies the regularity of the linear slicing moment $\mathbf{m}_d(y|u)$ for any given $u \in BV_c^+(\mathbb{R}^n)$.
2. The (generative) data model: Eqn. (18) specifies how the observable or computable data $\mathbf{m}_d(y|u_J)$ are generated from the unknown $\mathbf{m}_d(y|u)$.

In combination, they lead to our novel dejittering model built upon the Bayesian rationale [18], or equivalently in terms of the framework of inverse problems, the Tikhonov method [26].

3.3 Dejittering via moment regularization

For any fixed codimension d , we shall simplify the notations by defining $M(y) = M_d(y|u_J) = M_d(y|u)$, $\mathbf{m}_J(y) = \mathbf{m}_d(y|u_J)$, and $\mathbf{m}(y) = \mathbf{m}_d(y|u)$. For image and video dejittering, as in Eqn. (12), the jitter $s(y)$ is assumed to be of Gaussian type $\mathcal{N}(0^{n-d}, \Sigma)$ with a covariance matrix Σ . Also the data model in (18) reveals

$$s(y) = \frac{1}{M(y)} (\mathbf{m}_J(y) - \mathbf{m}(y)).$$

In combination with the BV regularity, the Bayesian/Tikhonov framework [9, 18], it leads to the following variational model for restoring the ideal linear moment $\mathbf{m}_d(y)$ from its jittered version $\mathbf{m}_J(y)$ (when $M \neq 0$):

$$\begin{aligned} & \min_{\mathbf{m}(y) \in BV_c(\mathbb{R}^d, \mathbb{R}^{n-d})} \int_{\mathbb{R}^d} |D\mathbf{m}(y)| \\ & + \frac{\lambda}{2} \int_{\mathbb{R}^d} \frac{1}{M^2(y)} (\mathbf{m}_J(y) - \mathbf{m}(y)) \Sigma^{-1} (\mathbf{m}_J(y) - \mathbf{m}(y))^T dy. \end{aligned} \quad (19)$$

The weight λ balances the regularity term and the fitting term, and the model is a regularized weighted least-square problem. In the fitting term, $M(y) = M_d(y|u_J)$ and $\mathbf{m}_J(y) = \mathbf{m}_d(y|u_J)$ are directly computable from a given jittered image $u_J(z)$, while $\mathbf{m}(y) = \mathbf{m}_d(y|u)$ is unknown. Furthermore, they satisfy the following compatibility condition. When $M = 0$, as it will be clear from the following Propositions, the fitting term goes to zero.

Proposition 3 (Compatibility Condition). *For any $u \in BV_c^+(\mathbb{R}^d)$, the condition $M(y) = M_d(y|u_J) = M_d(y|u) = 0$ implies that $\mathbf{m}_J(y) = 0$ and $\mathbf{m}(y) = 0$, for any $y \in \mathbb{R}^d$.*

Proof. $\forall y \in \mathbb{R}^d$, $M(y) = 0 \Leftrightarrow u(x, y) = 0$ for a.e. $x \in \mathbb{R}^{n-d}$, which implies that

$$\begin{aligned} \mathbf{m}_J(y) &= \mathbf{m}_d(y|u_J) = \int_{\mathbb{R}^{n-d}} xu(x - s(y), y) dx = 0, \\ \mathbf{m}(y) &= \mathbf{m}_d(y|u) = \int_{\mathbb{R}^{n-d}} xu(x, y) dx = 0. \quad \square \end{aligned}$$

Inspired by this proposition, we now study *independently* the properties of the dejittering energy

$$E[\mathbf{m}|\mathbf{m}_J, M] = \int_{\mathbb{R}^d} |D\mathbf{m}| + \frac{\lambda}{2} \int_{\mathbb{R}^d} \frac{1}{M^2} (\mathbf{m}_J - \mathbf{m}) \Sigma^{-1} (\mathbf{m}_J - \mathbf{m})^T dy, \quad (20)$$

for any given $\mathbf{m}_J$ and M , that are subject to:

- (A1) $M(y) \geq 0$, compactly supported, and $M \in L^\infty(\mathbb{R}^d)$;
- (A2) $M(y) = 0 \Rightarrow \mathbf{m}_J(y) = 0^{n-d}$, where $\mathbf{m}_J : \mathbb{R}^d \rightarrow \mathbb{R}^{n-d}$ is Lebesgue measurable; and
- (A3) $\mathbf{m}_J \in L^2(\mathbb{R}^d \rightarrow \mathbb{R}^{n-d}, d\mu)$, where $d\mu = \frac{1}{M^2} dy$ denotes the weighted measure on $\mathbb{R}^d$.

Proposition 4. *Let $\mathbf{m} \equiv 0^{n-d}$ be the zero vectorial function. Then, $E[\mathbf{m} = 0^{n-d}|\mathbf{m}_J, M] < \infty$.*

Proof. This is guaranteed by (A3), and the fact that

$$\mathbf{m}_J \Sigma^{-1} \mathbf{m}_J^T \leq \frac{1}{\lambda_{\min}(\Sigma)} |m_J|^2, \quad (21)$$

where $\lambda_{\min}(\Sigma) > 0$ denotes the smallest eigenvalue of Σ . $\square$

Proposition 5. *Suppose $\mathbf{m} \in BV(\mathbb{R}^d, \mathbb{R}^{n-d})$ and $E[\mathbf{m}|\mathbf{m}_J, M] < \infty$, then*

$$M(y) = 0 \text{ implies } \mathbf{m}(y) = 0^{n-d}, \text{ a.e. } y \in \mathbb{R}^d. \quad (22)$$

In particular, $\mathbf{m}(y)$ must be compactly supported and $\mathbf{m}(y) \in BV_c(\mathbb{R}^d, \mathbb{R}^{n-d})$.

Proof. By the assumption,

$$\int_{\mathbb{R}^d} \frac{1}{M^2} (\mathbf{m}_J - \mathbf{m}) \Sigma^{-1} (\mathbf{m}_J - \mathbf{m})^T dy < \infty.$$

Thus, $M = 0$ implies $(\mathbf{m}_J - \mathbf{m}) \Sigma^{-1} (\mathbf{m}_J - \mathbf{m})^T = 0$, for a.e. $y \in \mathbb{R}^d$. Since Σ is positive definite, this further implies $\mathbf{m}_J = \mathbf{m}$, a.e. in $\mathbb{R}^d$. Then, (22) follows directly from the assumption (A2) (or Proposition 3), and the compactness of M passes onto $\mathbf{m}$ as a result. $\square$

With these propositions, we now prove the existence and uniqueness of the minimizers to the dejittering energy (20).

Theorem 2. *Under the assumptions (A1), (A2) and (A3), the minimizer to energy $E[\mathbf{m}|\mathbf{m}_J, M]$ in (20) exists and is unique in $BV_c(\mathbb{R}^d, \mathbb{R}^{n-d})$.*

Proof. First, we prove the existence of the minimizer. By Proposition 4,

$$\inf_{\mathbf{m} \in BV_c(\mathbb{R}^d, \mathbb{R}^{n-d})} E[\mathbf{m}|\mathbf{m}_J, M] \leq E[0^{n-d}|\mathbf{m}_J, M] < \infty.$$

Let $\{\mathbf{m}^i(y)\}$ be a minimizing sequence in $BV_c(\mathbb{R}^d, \mathbb{R}^{n-d})$. Then, by Proposition 5, $\{\mathbf{m}^i(y)\}_{i=1}^\infty$ must be uniformly compactly supported, i.e., there exists a bounded open set U and a compact set $K \subseteq U$, such that

$$\text{supp } \mathbf{m}^i \subseteq K \subseteq U, \text{ for } i = 1, \dots, \infty. \quad (23)$$

In addition, by the assumption (A2) and Proposition 5, one can assume

$$\text{supp } \mathbf{m}_J, \text{ supp } M \subseteq K \subseteq U. \quad (24)$$

Then,

$$E[\mathbf{m}^i|\mathbf{m}_J, M] \equiv E[\mathbf{m}^i|\mathbf{m}_J, M, U], \quad (25)$$

where the latter refers to the energy restricted over U :

$$E[\mathbf{m}^i|\mathbf{m}_J, M, U] = \int_U |D\mathbf{m}^i| + \frac{\lambda}{2} \int_U \frac{1}{M^2} (\mathbf{m}_J - \mathbf{m}^i) \Sigma^{-1} (\mathbf{m}_J - \mathbf{m}^i)^T dy.$$

By the assumption (A1)

$$\frac{1}{M^2}(\mathbf{m}_J - \mathbf{m}^i)\Sigma^{-1}(\mathbf{m}_J - \mathbf{m}^i)^T \geq \frac{1}{\lambda_{\max}(\Sigma)} \frac{1}{\|M\|_{\infty}^2} |\mathbf{m}_J - \mathbf{m}^i|_2^2,$$

where $\lambda_{\max}(\Sigma)$ denotes the largest eigenvalue of the covariance matrix. Since $L^2(U, \mathbb{R}^{n-d}) \subseteq L^1(U, \mathbb{R}^{n-d})$ for any bounded domain U , the sequence $\{\mathbf{m}^i(y)|_U\}_{i=1}^{\infty}$ is a bounded sequence in $BV(U, \mathbb{R}^{n-d})$. Therefore, by the L^1 -weak compactness, there exists a subsequence $\{\mathbf{m}^k(y)|_U\} = \{\mathbf{m}^{i_k}(y)|_U\}$ that converges to some $\mathbf{m}^{\infty}$ in $L^1(U, \mathbb{R}^{n-d})$. One can further require that

$$\mathbf{m}^k(y) \longrightarrow \mathbf{m}^{\infty}(y), \text{ a.e. } y \in U. \quad (26)$$

Then, by the lower-semicontinuity property of the TV Radon measure under L^1 convergence,

$$\int_U |D\mathbf{m}^{\infty}| \leq \liminf_{k \rightarrow \infty} \int_U |D\mathbf{m}^k|. \quad (27)$$

On the other hand, by (26) and Fatou's Lemma:

$$\begin{aligned} \int_U \frac{1}{M^2}(\mathbf{m}_J - \mathbf{m}^{\infty})\Sigma^{-1}(\mathbf{m}_J - \mathbf{m}^{\infty})^T dy &\leq \\ \liminf_{k \rightarrow \infty} \int_U \frac{1}{M^2}(\mathbf{m}_J - \mathbf{m}^k)\Sigma^{-1}(\mathbf{m}_J - \mathbf{m}^k)^T dy. \end{aligned} \quad (28)$$

In combinations of (27), (28), and (25), we have

$$E[\mathbf{m}^{\infty}|\mathbf{m}_J, M, U] \leq \lim_{k \rightarrow \infty} E[\mathbf{m}^k|\mathbf{m}_J, M, U] = \lim_{k \rightarrow \infty} E[\mathbf{m}^k|\mathbf{m}_J, M].$$

By (23), one must have $\text{supp } \mathbf{m}^{\infty} \subseteq K \subseteq U$, and

$$E[\mathbf{m}^{\infty}|\mathbf{m}_J, M, U] = E[\mathbf{m}^{\infty}|\mathbf{m}_J, M].$$

Therefore, we have established

$$E[\mathbf{m}^{\infty}|\mathbf{m}_J, M] \leq \lim_{k \rightarrow \infty} E[\mathbf{m}^k|\mathbf{m}_J, M] = \inf_{\mathbf{m}} E[\mathbf{m}|\mathbf{m}_J, M].$$

Thus $\mathbf{m}^{\infty} \in BV_c(\mathbb{R}^d, \mathbb{R}^{n-d})$ has to be a minimizer.

Regarding the uniqueness, from the assumption (A1) on $M(y) \in L^{\infty}(\mathbb{R}^d)$, one has $M < \infty$ and $\frac{1}{M^2} > 0$ a.e. on $\mathbb{R}^d$. Then, it is trivial to see that $E[\mathbf{m}|\mathbf{m}_J, M]$ must be strictly convex in $BV_c(\mathbb{R}^d, \mathbb{R}^{n-d})$, and the minimizer has to be unique. $\square$

This theorem secures the feasibility of proper numerical computations of the proposed dejittering model. From the given image u_J , first compute the jittered linear moment $\mathbf{m}_J$, then apply the dejittering functional (20) to regularize this moment function. The regularized moment function m^* is then employed to estimate the unknown jitter $s(y)$. For the model and algorithm

to work effectively, one needs two pieces of input data: the jittered image $u_J \in \mathbb{R}^n$ and the statistics of the $(n-d)$ -jitter $s(y) \in \mathbb{R}^d$ (i.e., the covariance matrix Σ as modeled by (12), which is often obtained by suitable statistical estimators).

Algorithm:

1. Compute the marginal projection $M(y)$ and the linear slicing moment $\mathbf{m}_J(y)$ of image u_J .
2. Find the minimizer of (20), $\mathbf{m}^*(y) = \operatorname{argmin} E[\mathbf{m}|\mathbf{m}_J, M]$.
3. Compute the jitter by

$$s^*(y) = \begin{cases} \frac{\mathbf{m}_J - \mathbf{m}^*}{M}, & M(y) \neq 0 \\ 0^{n-d}, & M(y) = 0 \end{cases}.$$

4. Dejitter the image by s^* :

$$u^*(z) = u^*(x, y) = u_J(x + s^*(y), y).$$

In the next section, we discuss how to apply the above general framework to the practical application of 2-D image dejittering, for which $n = 2$, and $d = 1$.

4 Application to Image Dejittering and Examples

Let $\Omega_{R,H} = (-R, R) \times (0, H)$ denote a typical 2-D display domain, and an image defined on $\Omega_{R,H}$ be denoted by $v(x, y) \geq 0$ with $x \in (-R, R)$ and $y \in (0, H)$. A typical jitter can be modeled by a random map,

$$s = (0, H) \rightarrow \mathbb{R}, \quad y \rightarrow s(y).$$

As in Eqn. (12), assume that $s(y)$'s are i.i.d.'s of Gaussian type $\mathcal{N}(0, \sigma^2)$ with p.d.f.,

$$p(s(y) = a) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{a^2}{2\sigma^2}}, \quad \text{for any fixed } y.$$

Then, a jittered image v_J is defined as

$$v_J(x, y) = v(x - s(y), y) \geq 0. \quad (29)$$

In practice, both v_J and v are indeed only displayed or available on a finite domain $\Omega_{R,H}$. It is then necessary to specify the boundary filling mechanism when $|s(y)| \neq 0$. Depending on the situation, the filled-in data at the boundaries could be (i) random, (ii) generated by Neumann flat extension, or (iii) generated by other mechanisms such as symmetric extension. To avoid such complications, as well as to illustrate the application of the general theory developed above, we assume that the image domain is an ideal horizontal stripe

$\Omega_H = (-\infty, \infty) \times (0, H)$ (as in [23]) and that there exists some $R > 0$, such that

$$\text{supp}_{\Omega_H} v \subseteq \Omega_{R,H}, \quad \text{and } v \in BV^+(\Omega_H). \quad (30)$$

Then, v_J in (29) is always well-defined regardless of $s(y)$. Finally, by zero-padding, both v and v_J on Ω_H are extended to the entire plane $\mathbb{R}^2$, and denoted by u and u_J respectively. The jitter s is also naturally extended from $(0, H)$ to $\mathbb{R}^1$ by i.i.d. sampling. Then,

$$u_J(z) = u_J(x, y) = u(x - s(y), y), \quad \forall z = (x, y) \in \mathbb{R}^2,$$

and (30) implies that $u \in BV_c^+(\mathbb{R}^2)$.

Notice that $\int_{\mathbb{R}^2} |u(z)| dz = \int_{\Omega_H} |v(z)| dz$, and

$$\int_{\mathbb{R}^2} |D u| = \int_{\Omega_H} |D v| + \int_{\partial\Omega_H} |f_v| dH^1 < \infty,$$

where $\partial\Omega_H = (\mathbb{R}^1 \times \{0\}) \cup (\mathbb{R}^1 \times \{H\})$ denotes the lower and upper boundaries, dH^1 the 1-dimensional Hausdorff measure, and $f_v = \text{Tr}(v)$ the trace of v along $\partial\Omega_H$ [13].

Thus, we are able to apply the general framework in the previous sections for the dejittering of u_J (and consequently for v_J). Define accordingly,

$$m(y) = \int_{\mathbb{R}} x u(x, y) dx, \quad m_J(y) = \int_{\mathbb{R}} x u_J(x, y) dx, \quad \text{and}$$

$$M(y) = \int_{\mathbb{R}} u_J(x, y) dx (= \int_{\mathbb{R}} u(x, y) dx).$$

The dejittering model (20) becomes to minimize

$$E[m|m_J, M] = \int_{\mathbb{R}} |D m| + \frac{\mu}{2} \int_{\mathbb{R}} \frac{1}{M^2} (m_J - m)^2 dy, \quad (31)$$

where $\mu = \frac{\lambda}{\sigma^2}$. Eqn. (31) is a regularized weighted (by M^{-2}) least-square problem. If M were a constant, this equation would become precisely the 1-D version of the celebrated TV restoration model of Rudin-Osher-Fatemi [21].

For most digital devices, one has $u \in [0, 1]$ or $[0, 255]$ (8-bit). Then, the compactness of u ensures $M \in L^\infty(\mathbb{R})$. As long as $m_J \in L^2(\mathbb{R}, \frac{1}{M^2} dy)$, all the three conditions (A1), (A2) and (A3) of Theorem 2 are naturally satisfied. The optimal estimator $m^* = \text{argmin } E[m|m_J, M]$ therefore must exist uniquely.

In terms of numerical computations, there have been quite a few effective methods in the literature for models like (31), e.g., [3, 4, 21, 26]. One frequently adopted approach is based upon the formal Euler-Lagrange equation of (31),

$$D \left[\frac{D m(y)}{|D m(y)|} \right] + \frac{\mu}{M^2(y)} (m_J(y) - m(y)) = 0, \quad (32)$$

or equivalently,

$$M^2(y)D \left[\frac{D m(y)}{|D m(y)|} \right] + \mu(m_J(y) - m(y)) = 0, \quad (33)$$

where $D = d/dy$ is the derivative along the codimension y . It is evident from the last equation that $M(y) = 0$ implies $m(y) = m_J(y)$, which further leads to $m(y) = 0$ because of the assumption (A2) in Theorem 2. As common in the literature [7, 21, 26], a regularization parameter $\epsilon > 0$ can be introduced to replace $|Dm(y)|$ in the denominator by $|Dm|_\epsilon = \sqrt{\epsilon^2 + |Dm|^2}$ in (33). The nonlinear equation (33) can be solved iteratively by the lagged diffusivity fix-point method as in Acar and Vogel [1]. We refer to the remarkable monograph of Vogel for more details on the effective computations of models like (31)-(33), including discussions on the selection of the weighting parameter μ .

Numerical Examples

Finally, we demonstrate the computational performance of the new dejittering model through some typical examples. Notice that our model naturally applies to color images as well [2, 5]. The following two results, Fig. 3 and Fig. 4, are in color; see Figs. A.10 and A.11 in the ‘Color Figures’ appendix.

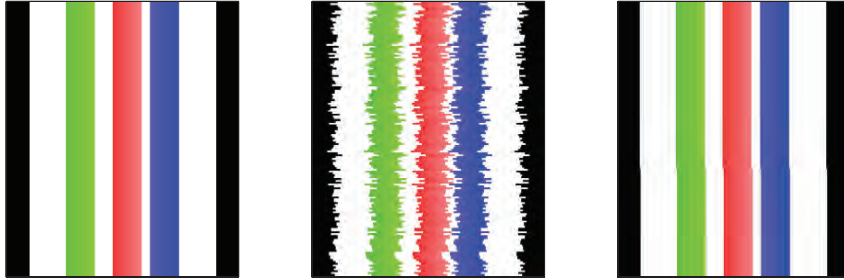


Fig. 3. (a) Ideal image u . (b) Jittered image u_J . (c) Dejittered image u^* via moment regularization. (Color images in Figure A.10.)

The first example in Fig. 3 shows a synthetic piecewise constant image u , its jittered version u_J , and the dejittered image u^* via our new model based upon moment regularization. Since most images in the real world are often noisy, in Fig. 4 we have tested the robustness of our new model in the presence of intensity noises. The dejittered image in (c) clearly confirms such robustness, thanks to the averaging (or lowpass filtering) nature of moment integrals. In Fig. 5 and Fig. 6, via a standard test image in image processing, we have explicitly demonstrated the moment sequence from our dejittering model: the ideal moment $m(y)$, the jittered moment $m_J(y)$, and the optimally

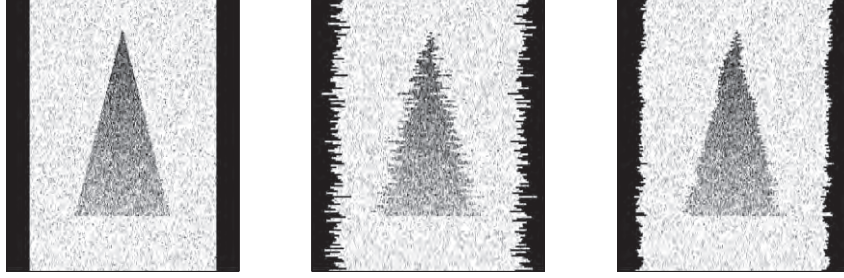


Fig. 4. Ideal image u is with intensity Gaussian white noise three vertical bars. (a) Original image u , (b) Jittered image u_J . (c) Dejittered image. The dejittered estimation in (c) shows the robustness of our model to the perturbation of intensity noises. (Color images in Figure A.11.)

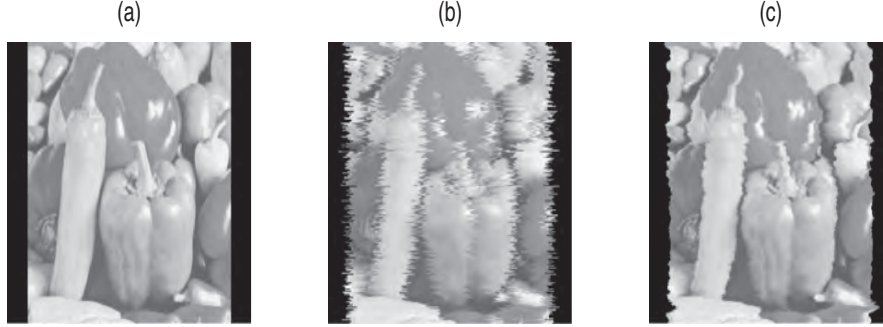


Fig. 5. Dejittering a standard test image of peppers via moment regularization.

estimated moment $m^*(y)$. Finally, Fig. 7 shows the performance of the model on another standard test image of “Barbara” in image processing.

We have compared this method with the Bake-and-Shake dejittering method in [14]. In Fig. 8, (c) shows an example of using the method in [14] and (d) shows the result from the current approach. By comparison, the main body of the boat is better restored by the Bake-and-Shake methods. However, the moment approach has reduced the dimension of the problem and computation is much faster. In addition, for the *thin* masts of the boat which do not have good spatial correlations after jittering, the current approach based on moment regularization seems to achieve better performance. Motivated by this example, in Fig. 9, we have combined the two methods by further applying the Bake-and-Shake algorithm to the output from the moment regularization approach, i.e., the image (d) in Fig. 9. The final quality of dejittering is noticeably improved.

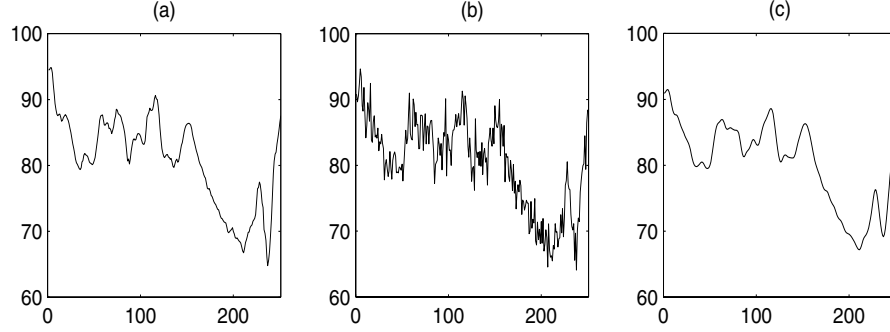


Fig. 6. The associated moment profiles corresponding to the images in Fig. 5.



Fig. 7. The performance of the new model on the standard test image of “Barbara.” Dejittering images with rich textures has been a challenging task for PDE (or diffusion) based methods [14, 23].

5 Conclusion

Motivated by the image dejittering problem in contemporary imaging science, the current paper introduces the notion of slicing moments of BV functions (or images), and studies their mathematical properties and regularization techniques. Under the Bayesian rationale for general restoration problems, the regularities of the slicing moments lead to a variational dejittering model that involves weighted least-square optimization and the total variation Radon measure. The existence and uniqueness of the optimal solutions, as well as the associated computational approaches are all explored under the most general settings and assumptions. In practice, our novel dejittering model introduces dimensionality reduction and gains remarkable computational efficiency.

Our future work will focus on improving the model to achieve maximal degrees of accuracy, performance, and computational efficiency.

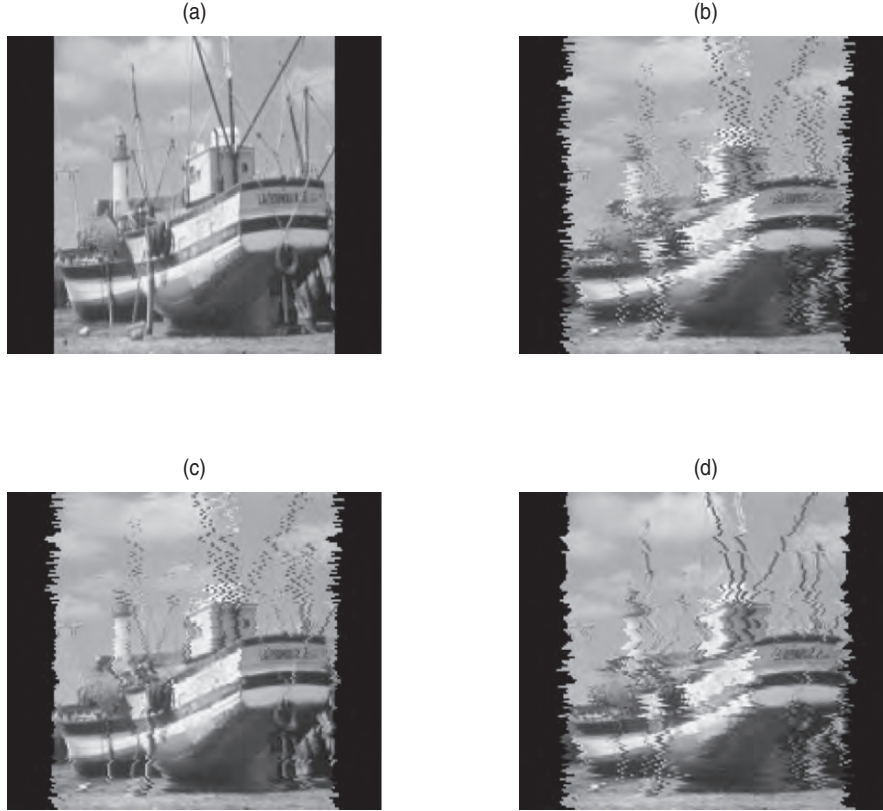


Fig. 8. (a) Original image. (b) Jittered image. (c) Dejittered image by the Bake-and-Shake method in [14]. (d) Dejittered image by moment regularization. The moment regularization yields better restoration for thin features, while not robust where moment information is weak.

Acknowledgement. We thank Stan Osher and Tony Chan for their constant inspirations and encouragement on this emerging field of mathematical image and vision analysis.

References

1. R. Acar and C. R. Vogel. Analysis of total variation penalty methods for ill-posed problems. *Inverse Prob.*, 10:1217–1229, 1994.
2. J.-F. Aujol and S.-H. Kang. Color image decomposition and restoration. *Journal of Visual Communication and Image Representation (in press)*, 2005.
3. A. Chambolle and P. L. Lions. Image recovery via Total Variational minimization and related problems. *Numer. Math.*, 76:167–188, 1997.



Fig. 9. (a) The jittered image in (b) of Fig. 8. (d) Dejittered image via combining the Bake-and-Shake method and moment regularization.

4. T. F. Chan, G. H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 20:1964–1977, 1999.
5. T. F. Chan, S.-H. Kang, and J. Shen. Total variation denoising and enhancement of color images based on the CB and HSV color models. *J. Visual Comm. Image Rep.*, 12(4):422–435, 2001.
6. T. F. Chan, S. Osher, and J. Shen. The digital TV filter and non-linear denoising. *IEEE Trans. Image Process.*, 10(2):231–241, 2001.
7. T. F. Chan and J. Shen. *Image Processing and Analysis: variational, PDE, wavelets, and stochastic methods*. SIAM Publisher, Philadelphia, 2005.
8. T. F. Chan and J. Shen. Variational image inpainting. *Comm. Pure Applied Math.*, 58:579–619, 2005.
9. T. F. Chan, J. Shen, and L. Vese. Variational PDE models in image processing. *Notices Amer. Math. Soc.*, 50:14–26, 2003.
10. I. Daubechies. *Ten lectures on wavelets*. SIAM, Philadelphia, 1992.
11. G. B. Folland. *Real Analysis - Modern Techniques and Their Applications*. John Wiley & Sons, Inc., second edition, 1999.
12. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6:721–741, 1984.
13. E. Giusti. *Minimal Surfaces and Functions of Bounded Variation*. Birkhäuser, Boston, 1984.
14. S.-H. Kang and J. Shen. Video dejittering by bake and shake. *Image Vis. Comput.*, 24(2):143–152, 2006.
15. A. Kokaram and P. Rayner. An algorithm for line registration of TV images based on a 2-D AR model. *Signal Processing VI, Theories and Applications*, pages 1283–1286, 1992.
16. A. Kokaram, P. M. B. Roosmalen, P. Rayner, and J. Biemond. Line registration of jittered video. *IEEE Int'l Conference on Acoustics, Speech, and Signal Processing*, pages 2553–2556, 1997.
17. Y. Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*, volume 22 of *University Lecture Series*. AMS, Providence, 2001.

18. D. Mumford. *Geometry Driven Diffusion in Computer Vision*, chapter “The Bayesian rationale for energy functionals”, pages 141–153. Kluwer Academic, 1994.
19. D. Mumford. Pattern theory: The mathematics of perception. *Int’l Congress Mathematicians (ICM)*, III, Beijing, 2002.
20. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Applied. Math.*, 42:577–685, 1989.
21. L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
22. J. Shen. On the foundations of vision modeling I. Weber’s law and Weberized TV restoration. *Physica D: Nonlinear Phenomena*, 175:241–251, 2003.
23. J. Shen. Bayesian video dejittering by BV image model. *SIAM J. Appl. Math.*, 64(5):1691–1708, 2004.
24. L. A. Vese. A study in the BV space of a denoising-deblurring variational problem. *Appl. Math. Optim.*, 44(2):131–161, 2001.
25. L. A. Vese and S. J. Osher. Modeling textures with Total Variation minimization and oscillating patterns in image processing. *J. Sci. Comput.*, 19(1-3):553–572, 2003.
26. C. Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, 2002.
27. J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, Stuttgart, Germany, 1998.
28. S. C. Zhu, Y. N. Wu, and D. Mumford. Minimax entropy principle and its applications to texture modeling. *Neural Comput.*, 9:1627–1660, 1997.

CLG Method for Optical Flow Estimation Based on Gradient Constancy Assumption

Adam Rabcewicz

Faculty of Mathematics and Computer Science, Nicolaus Copernicus University,
Toruń, Poland. E-mail: `adrab@mat.uni.torun.pl`

Summary. Many differential methods for optical flow computation are extensions of the Lucas-Kanade technique or the Horn-Schunck approach. Both exploit a brightness constancy assumption. The former method is local and it is recognized for its robustness under noise. The latter one is global and it yields a dense flow field. Recently Bruhn et al. have introduced the so-called combined local-global (CLG) method which incorporates advantages of both techniques.

We propose a modification of the CLG method which consists in replacing the brightness constancy assumption with constancy of the gradient along trajectories. This leads to an energy functional which has essentially the same structure as in the original CLG approach. The modified method gives considerably smaller angular errors for sequences with domination of translatory motions.

Key words: Optical flow, motion estimation, variational techniques, differential methods.

1 Introduction

Differential methods belong to the most successful techniques for computing optical flow in image sequences. Many of them are based on the Optical Flow Constraint (OFC), which is derived from the Brightness Constancy Assumption (BCA). Bruhn et al. [3] classify them into two categories: *local* and *global*, according to the way that they cope with the *aperture problem*. Local methods, as for example the classical Lucas and Kanade approach [8], exploit the information in some neighborhood of the estimated location. These methods are known for their robustness under noise but they give nondense flow fields. Global methods, on the other hand, yield optical flow fields with 100% density but they are more sensitive against noise. This category includes another classical approach by Horn and Schunck [7], where the aperture problem is solved by adding a smoothness constraint to the OFC. These classical methods admit many extensions; for more details we refer the reader to [3] and to the references given there.

The so-called combined local and global (CLG) method was proposed by Bruhn et al. [3]. It combines ideas and advantages of both approaches discussed above. It is local and global at the same time. It is robust against noise and gives a dense optical flow field.

The BCA means that intensity values of corresponding points are invariant while moving from frame to frame. It is not the case in most natural scenes. Therefore the BCA is often supported (or even replaced) by another constraint e.g. Gradient Constancy Assumption (GCA) [6, 10, 11, 12, 2]. This model permits linear variations of brightness but, as pointed in Barron et al. [1], it implies that rigid deformations (as rotations or dilations) should not be present. So, it is not surprising that this assumption is particularly useful for sequences where translatory motion dominates.

In this contribution we propose a modification of the CLG method which we denote by CLG(H). It consists in replacing the BCA with the GCA. It is shown that this leads to an energy functional which has essentially the same structure as in the original CLG approach. As a consequence, numerical schemes remain almost unchanged. Thus, without much work, a considerable accuracy improvement of estimated flows for some sequences is reported.

Some related work. Early works exploiting the GCA used it to overcome the aperture problem by adding supplementary equations to the OFC [6, 10]. Obtained this way an overdetermined system of equations has been solved via the pseudo-inverse formalism. Uras et al. [11] proceeded similarly but they used only the GCA without the OFC. All mentioned methods can be classified as local. Embedding the CGA into a global energy functional was suggested by Weickert et al. [12]. In fact, they analyzed various data terms and smoothness constraints in a variational framework. Brox et al. [2] exploit both the BCA and the GCA in a nonlinearized form. They invented very effective numerical scheme, which provides some of the best results known so far.

Our proposed technique differs from the above methods by the fact that it simultaneously uses global and local conditions concerning the GCA. It is also worth to emphasize that the spatiotemporal constancy of the gradient was used only in [6] so far. Others mentioned techniques assume that spatial gradient remains unchanged while moving.

2 Review of the CLG Method

We follow the notation used in [3]. Thus, let $f: \Omega \times [0, T] \rightarrow \mathbb{R}$ denote an image sequence (it can be Gaussian presmoothed), where $\Omega \subset \mathbb{R}^2$ is a rectangular domain. Fix some $t \in [0, T]$. We want to determine the displacement vector field $\mathbf{w}(t) = (u(t), v(t), 1)^T$, $u(t), v(t): \Omega \rightarrow \mathbb{R}$, which matches objects in subsequent frames at times t and $t + 1$. The BCA means that the intensity $f(x, y, t)$ of a pixel at location $(x, y) \in \Omega$ at time $t \in [0, T]$ does not change

along trajectory $t \rightarrow (x(t), y(t), t)$, i.e.:

$$\frac{df(x(t), y(t), t)}{dt} = 0. \quad (1)$$

Applying the chain rule to (1) we obtain the OFC:

$$\nabla_3 f^T \mathbf{w} = 0, \quad (2)$$

where $\nabla_3 f = (f_x, f_y, f_t)^T$ denotes the spatiotemporal gradient of f and

$$(u, v)^T = \left(\frac{dx}{dt}, \frac{dy}{dt} \right)^T \quad (3)$$

is the optical flow vector. Unfortunately, scalar Eq. (2) is not sufficient for finding both components of the optical flow. This problem is known as the *aperture problem*. Eq. (2) determines only so-called *normal flow* i.e. vector component in the direction of the spatial gradient of the image $\nabla f = (f_x, f_y)^T$. Therefore the OFC has to be supplemented by additional assumptions.

Lucas and Kanade [8] coped with the aperture problem by an assumption that the velocity field is constant within some neighborhood of size ρ . This constraint can be formulated as a problem of minimizing the quadratic form

$$E_{LK}(\mathbf{w}) = \mathbf{w}^T J_\rho \mathbf{w}, \quad (4)$$

where $J_\rho := K_\rho * (\nabla_3 f \nabla_3 f^T)$ is the structure tensor, K_ρ is a 2D Gaussian kernel with standard deviation ρ and the symbol $*$ denotes the convolution in each matrix component (note that with this notation we have $(\nabla_3 f^T \mathbf{w})^2 = \mathbf{w}^T J_0 \mathbf{w}$). The minimizer of (4) is the solution of 2×2 system of linear equations for the two unknowns u and v . The velocity vector can be estimated only at locations where the system matrix is invertible. Therefore, the resulting field is nondense.

Another classical method proposed by Horn and Schunck [7] solves the aperture problem by regularization of the velocity field. It determines the optic flow by minimizing the global energy functional

$$E_{HS}(\mathbf{w}) = \int_{\Omega} (\mathbf{w}^T J_0 \mathbf{w} + \alpha |\nabla \mathbf{w}|^2) dx dy, \quad (5)$$

where $|\nabla \mathbf{w}|^2 = |\nabla u|^2 + |\nabla v|^2$ and $\alpha > 0$ is a regularization parameter. Thanks to the regularizer $|\nabla \mathbf{w}|^2$ it is possible to determine the velocity vector at all locations (it fills in the information from the neighborhood, if necessary). Thus, the resulting flow field benefits from 100% density. On the other hand, this method is more sensitive under noise than the previous one.

Recently, Bruhn et al. [3] extended the Horn and Schunck technique by replacing the matrix J_0 with the structure tensor J_ρ . They combined the above techniques by considering the CLG functional

$$E_{CLG}(\mathbf{w}) = \int_{\Omega} (\mathbf{w}^T J_{\rho} \mathbf{w} + \alpha |\nabla \mathbf{w}|^2) dx dy. \quad (6)$$

A nonlinear variant of (6) was also considered in order to make both terms more robust against outliers:

$$E_{CLG-N}(\mathbf{w}) = \int_{\Omega} (\psi_1(\mathbf{w}^T J_{\rho} \mathbf{w}) + \alpha \psi_2(|\nabla \mathbf{w}|^2)) dx dy, \quad (7)$$

where $\psi_i: \mathbb{R} \rightarrow \mathbb{R}$, $i = 1, 2$ are non-quadratic penalizers. They used the function proposed by Charbonnier et al. [4]:

$$\psi_i(s^2) = 2\beta_i^2 \sqrt{1 + \frac{s^2}{\beta_i^2}}, \quad i = 1, 2, \quad (8)$$

where β_i are scaling parameters.

The nonlinear variant of the CLG approach has another extension based on the multiresolution technique. It improves the resulting flow field by avoiding the linearization of the BCA. A coarse-to-fine strategy is applied, i.e. the motion increment $\delta \mathbf{w}^m$ at level m is obtained by minimization of the following functional:

$$E_{CLG-N}^m(\delta \mathbf{w}^m) = \int_{\Omega} (\psi_1(\delta \mathbf{w}^{mT} J_{\rho}^m \delta \mathbf{w}^m) + \alpha \psi_2(|\nabla(\mathbf{w}^m + \delta \mathbf{w}^m)|^2)) dx dy. \quad (9)$$

Here J_{ρ}^m is the structure tensor of warped original sequence by the optical flow at level m , which is the sum of the motion increments at coarser scales: $\mathbf{w}^m := \mathbf{w}^{m-1} + \delta \mathbf{w}^{m-1}$. The procedure starts from $m = 0$ (the coarsest level) and $\mathbf{w}^0 = (0, 0, 0)$.

All variants of the CLG methods (linear, nonlinear and multiresolution) have spatiotemporal equivalents. Formally, they simply consist in replacing Ω with $\Omega \times [0, T]$ under the integral, replacing the spatial smoothness term $|\nabla \mathbf{w}|^2$ with the spatiotemporal regularizer $|\nabla_3 \mathbf{w}|^2$ and treating K_{ρ} in the structure tensor J_{ρ} as 3D Gaussian kernel. In this case the spatiotemporal presmoothing of input sequence is also applied.

3 Formulation of the CLG(H) Method

We demand spatiotemporal constancy of the gradient along the trajectory, that is,

$$\frac{d\nabla_3 f(x(t), y(t), t)}{dt} = 0. \quad (10)$$

Linearization of (10) leads to the following equations:

$$\begin{cases} (\nabla_3 f_x)^T \mathbf{w} = 0, \\ (\nabla_3 f_y)^T \mathbf{w} = 0, \\ (\nabla_3 f_t)^T \mathbf{w} = 0. \end{cases} \quad (11)$$

We embed this into the variational framework:

$$E_H(\mathbf{w}) = \int_{\Omega} (((\nabla_3 f_x)^T \mathbf{w})^2 + ((\nabla_3 f_y)^T \mathbf{w})^2 + ((\nabla_3 f_t)^T \mathbf{w})^2 + \alpha |\nabla \mathbf{w}|^2) dx dy, \quad (12)$$

But

$$((\nabla_3 f_x)^T \mathbf{w})^2 + ((\nabla_3 f_y)^T \mathbf{w})^2 + ((\nabla_3 f_t)^T \mathbf{w})^2 = \mathbf{w}^T H^2 \mathbf{w}, \quad (13)$$

where H denotes the Hessian matrix of f . So, if we want to obtain a CLG-like functional, we should write

$$E_{CLG(H)}(\mathbf{w}) = \int_{\Omega} (\mathbf{w}^T H_{\rho}^2 \mathbf{w} + \alpha |\nabla \mathbf{w}|^2) dx dy, \quad (14)$$

where $H_{\rho}^2 = (K_{\rho} * H)^2$. This way we impose a local assumption concerning the constancy of the gradient in some neighborhood of the estimated location.

One should note that the CLG(H) functional differs from the CLG only by the matrix in the data term. Formally, the structure tensor J_{ρ} has been replaced with the squared smoothed Hessian H_{ρ}^2 . In consequence, obtaining the energy functionals for both the nonlinear and the multiresolution variants of the CLG(H) method is straightforward and we skip it.

4 Algorithmic Realization

Algorithms used in [3, Section 6] for all variants of the original CLG method have been obtained by discretization of the Euler–Lagrange equations corresponding to suitable energy functionals using standard finite differences schemes. The resulting sparse linear system of equations was solved iteratively by the successive over-relaxation (SOR) method. These algorithms can be easily adopted for linear and nonlinear variants of the CLG(H) method – it is sufficient to change components J_{nm} of J_{ρ} to components H_{nm} of H_{ρ}^2 . They have been approximated using either the stencil $(1, -2, 1)$ or $(-1, 16, -30, 16, -1)/12$ for second-order x - and y -derivatives and only the former for the t -derivative. All mixed second-order derivatives have been computed with the stencil:

1	0	-1
0	0	0
-1	0	1

As a consequence, three consecutive frames are needed for 2D variants of the CLG(H) method to estimate optical flow at fixed time t .

The cost of computation is nearly the same as in case of the CLG method, apart from the first iteration, where, in case of the CLG(H), we have to compute the square of the smoothed Hessian.

The multiresolution variant needs additional explanations. In case of the 2D variant of the CLG method, after computing the motion increment $\delta \mathbf{w}^m$

at level m , the whole original sequence is warped by means of a backward registration, which is then used to determine the data term at next level. Our approach is slightly different. First of all, we do not proceed with the whole sequence, we are only interested in the optical flow at fixed time t . Moreover, at each level m we use not only warped frames but also the original one. More precisely, suppose we work with frames at times $t-1$, t , $t+1$ and that we have already computed the motion increment $\delta \mathbf{w}^m$ at level m . Then we warp the frame at time $t-1$ by motion field $\mathbf{w}^{m+1}$, and frame at time $t+1$ by $-\mathbf{w}^{m+1}$. Warped frames and the original one at time t are then used to recompute second-order derivatives in the data term at level $m+1$.

In our experiments, values for scaling parameters β_i within the penalizing functions have been optimized.

5 Comparison Between Methods

First we provide qualitative comparison on the well known *Yosemite* sequence with clouds¹. This sequence combines the divergent motion of mountains with translational motion of the sky. We have chosen it because of linear changes in the intensity of clouds, so we expect that the CLG(H) method gives better result there than the CLG. Actually it is so, as Figure 1 shows.

In Figure 2 we have juxtaposed the ground truth flow field with the result of the 2D multiresolution variant of our approach. As we can see, they match perfectly. The translational motion of the sky has been estimated correctly, similarly to the linear variant, but in this case, the discontinuities of motion have been preserved due to penalizer in smoothness term.

The quantitative comparison is made in Table 1. Efficiency of methods is expressed by the Average Angular Error (AAEs) between the estimated flow and the ground truth flow. Table 1 concerns sequences for which the AAEs were computed for the CLG technique in [3], i.e.: the *Yosemite* sequences with and without clouds², the *Office*³ and the *Marble*⁴ sequences. We gather here AAEs of 2D variants of the CLG and the CLG(H) methods for these sequences. The qualitative superiority of the CLG(H) method for the *Yosemite* sequence with clouds is confirmed quantitatively. It is worth to emphasize that the AAE result of the 2D multiresolution variant of our method for this sequence belongs to the best among all 2D results from the literature.

The significant improvement of the AAE is seen also for the *Marble* sequence, where only the translational motion appears. In remaining sequences

¹Created by Lynn Quam.

²The modified variant of *Yosemite* sequence without clouds is available from <http://www.cs.brown.edu/people/black/images.html>.

³Created by Galvin et al. [5], available from <http://www.cs.otago.ac.nz/research/vision>.

⁴Created by Otte and Nagel [9], available from http://i21www.ira.uk.de/image_sequences.

(i.e. the *Yosemite* without clouds and the *Office*) the divergent motion is dominating, but as we can see, the CLG(H) copes quite well with them. In spite of worse results for the linear variant, the CLG(H) finally outperforms the CLG.

Robustness under Gaussian noise is one of the main advantages of the CLG approach. It has been also examined for the CLG(H) method. Results of its 2D linear variant for the *Yosemite* sequence with clouds are shown in Table 2. As we can see, the CLG(H) method is, in general, much more sensitive to noise than the CLG (we get similar result only for small noise level). It is not surprising because the noise distorts the estimation of second-order derivatives much more than the first-order ones. But the situation changes completely when spatiotemporal presmoothing is applied to the noisy sequence instead of spatial only. This can be seen in Table 3. Spatiotemporal prefiltering improves significantly the reconstruction of noisy data and, in this case, the 2D variant of CLG(H) gives considerably better results than the 3D variant of the CLG!

Table 4 shows the results of investigation of the 2D multiresolution variant of the CLG(H) method with respect to parameter variations. As we can see, deviations from the optimum by factor two hardly influence the AAE. So, it can be stated that our method should work well in practice, when the parameters are not set to optimal.

Table 1. AAE for 2D linear, nonlinear and multiresolution variants of CLG and CLG(H) methods using various sequences.

Sequence	Linear		Nonlinear		Multiresolution	
	CLG	CLG(H)	CLG	CLG(H)	CLG	CLG(H)
Yosemite with clouds	7.14°	5.55°	6.03°	3.42°	4.86°	2.28°
Yosemite without clouds	2.64°	2.97°	2.31°	2.60°	1.62°	1.53°
Office	4.33°	4.60°	4.13°	3.75°	-	-
Marble	5.30°	3.14°	5.14°	2.59°	-	-

Table 2. Robustness under noise of 2D CLG and CLG(H) methods for the *Yosemite* sequence with clouds. Gaussian noise was added with zero mean and different standard deviation σ_n .

σ_n	2D CLG	2D CLG(H)
0	7.14° ± 9.28°	5.55° ± 8.63°
10	9.19° ± 9.62°	9.16° ± 9.66°
20	10.17° ± 10.50°	13.30° ± 9.98°
40	15.82° ± 11.53°	18.81° ± 12.28°

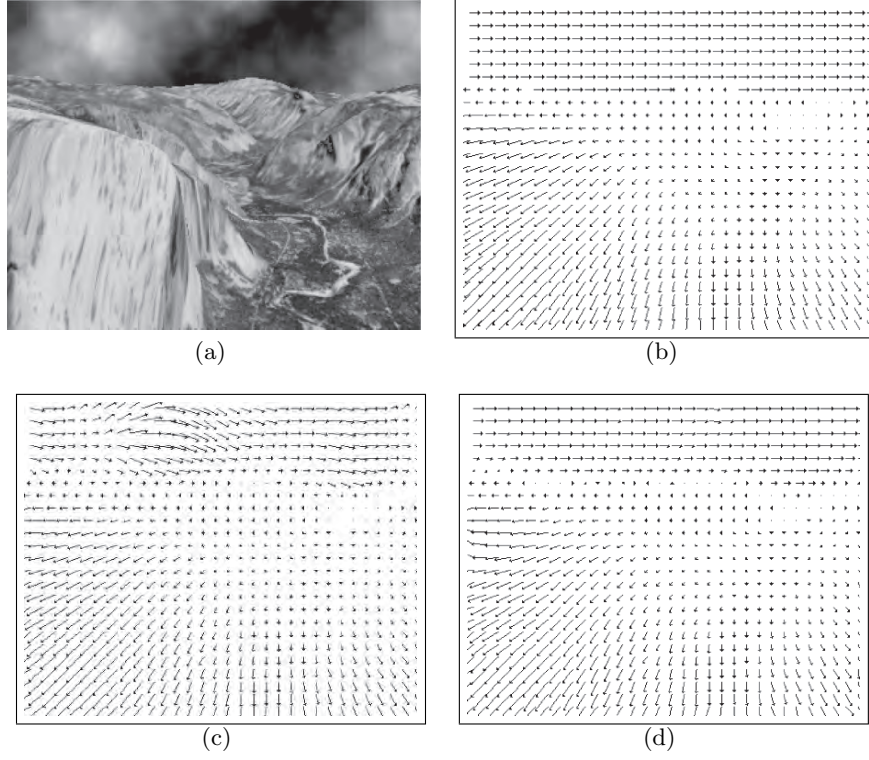


Fig. 1. (a) Frame 8 of the *Yosemite* sequence. (b) Ground truth flow field. (c) Computed flow field using 2D linear variant of the CLG method. (d) Computed flow field using 2D linear variant of the CLG(H) method.

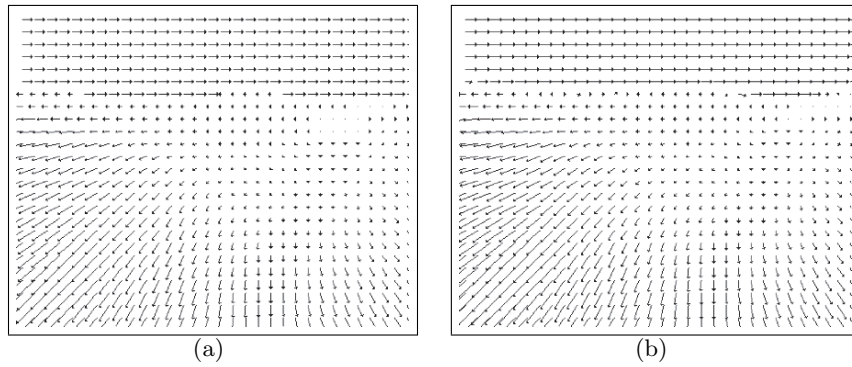


Fig. 2. (a) Ground truth flow field. (b) Computed flow field using 2D multiresolution variant of CLG(H) method.

Table 3. Robustness under noise of the 3D CLG method and the 2D CLG(H) method with spatiotemporal presmoothing for the *Yosemite* sequence with clouds. Gaussian noise was added with zero mean and different standard deviation σ_n .

σ_n	3D CLG	2D CLG(H) STP
0	$6.18^\circ \pm 9.19^\circ$	$4.69^\circ \pm 8.66^\circ$
10	$7.25^\circ \pm 9.39^\circ$	$5.89^\circ \pm 8.25^\circ$
20	$8.62^\circ \pm 9.97^\circ$	$7.66^\circ \pm 8.52^\circ$
40	$11.21^\circ \pm 11.19^\circ$	$10.31^\circ \pm 9.40^\circ$

Table 4. Stability of the CLG(H) method under parameter variations. Results for the *Yosemite* sequence with clouds.

σ	α	ρ	AAE
0.6	620	0.7	2.28°
0.3	"	"	2.35°
1.2	"	"	2.45°
0.6	620	0.7	2.28°
"	310	"	2.44°
"	1240	"	2.77°
0.6	620	0.7	2.28°
"	"	0.35	2.43°
"	"	1.4	2.51°

6 Summary

In this contribution we have proposed a CLG version with the gradient constancy assumption as matching criterion. Such a modification leads to minimization of energy functional, which is very similar to the one occurring in the CLG. Numerical schemes for linear and nonlinear variants of our method remains almost unchanged while experiments shows that our approach yields much better results for some sequences.

Most promising is 2D multiscale strategy and in our future work this algorithm will be parallelized.

References

1. J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *Int. J. Comput. Vis.*, 12(1):43–77, 1994.
2. T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Proceedings of the 8th European Conference on Computer Vision*, volume 3024 of *Lecture Notes in Computer Science*, pages 25–36, Prague, Czech Republic, 2004. Springer.

3. A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *Int. J. Comput. Vis.*, 61(3):211–231, 2005.
4. P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proc. 1994 IEEE International Conference on Image Processing*, volume 2, pages 168–172, Austin TX, 1994. IEEE Computer Society Press.
5. B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: An analysis of eight flow algorithms. In *Proc. 1998 British Machine Vision Conference*, Southampton, England, 1998.
6. R.M. Haralick and J.S. Lee. The facet approach to optic flow. In L.S. Baumann, editor, *Proceedings Image Understanding Workshop*, pages 84–93, Arlington, Va, 1983. Science Applications.
7. B. Horn and B. Schunck. Determining optical flow. *Artif. Intell.*, 17:185–203, 1981.
8. B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, Canada, 1981.
9. M. Otte and H.-H. Nagel. Estimation of optical flow based on higher-order spatiotemporal derivatives in interlaced and non-interlaced image sequences. *Artif. Intell.*, 78:5–43, 1995.
10. O. Tretiak and L. Pastor. Velocity estimation from image sequences with second order differential operators. In *Proceedings of the International Conference on Pattern Recognition*, pages 16–19, Montreal, Que, 1984.
11. S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biol. Cybern.*, 60:79–87, 1988.
12. J. Weickert, A. Bruhn, N. Papenberg, and T. Brox. Variational optic flow computation: From continuous models to algorithms. In L. Alvarez, editor, *IWCVIA'03: International Workshop on Computer Vision and Image Analysis*, volume 0026 of Cuadernos del Instituto Universitario de Ciencias y Tecnologías Cibernéticas, pages 1–6, Feb. 2004.

Denoising and Total Variation Methods

On Multigrids for Solving a Class of Improved Total Variation Based Staircasing Reduction Models

Joseph Savage and Ke Chen

Department of Mathematical Sciences, University of Liverpool, Peach Street,
Liverpool L69 7ZL, United Kingdom.

E-mail: k.chen@liverpool.ac.uk, url: <http://www.liv.ac.uk/~cmchenke>

Summary. Total variation regularization is well established as a denoising procedure with excellent edge capturing properties, however images denoised using total variation regularization suffer from the staircasing effect. Many models to reduce this effect have been proposed in the literature but not all models can be solved effectively. Our interest is in the fast iterative solution of the nonlinear partial differential equations arising from these models, specifically the use of nonlinear multigrid methods.

In this paper we first survey a class of staircasing reduction models and then focus on using effective solution as a criterion to find the most suitable model in this class of models that maintains edges by compromising in some way between Total Variation and H^1 regularization. We then compare the performance of nonlinear multigrid solvers, the fixed point iteration method using linear multigrid inner solvers and the explicit time marching (gradient descent) approaches.

Key words: Image restoration, denoising, regularization, nonlinear solvers, multilevel methods, staircasing reduction.

1 Introduction

During recording and transmission an image will often become contaminated with random Gaussian type noise; this is modeled by the equation

$$z(x, y) = u(x, y) + n(x, y), \quad x, y \in \Omega$$

where Ω is a bounded and open domain of $\mathbb{R}^2$ (usually a rectangle). Here z is a real function representing the observed (known) image, which in practice will be a discrete quantity (given in the form of $n \times m$ pixel values), u is the true image (unknown) and n is an additive (unknown) noise term. The problem of recovering u from z is an ill-conditioned inverse problem.

Image denoising methods use regularization techniques based on a priori knowledge of the image properties to approximate u . An early approach was H^1 regularization given by the following minimization problem:

$$\min_u J_{H^1}(u), \quad J_{H^1}(u) = \int_{\Omega} \alpha |\nabla u|^2 + \frac{1}{2}(u - z)^2 dx dy$$

The convex functional $J_{H^1}(u)$ is made up of a regularization functional $\int |\nabla u|^2$, which penalizes against non-smooth images, and a fit to data functional $\int \frac{1}{2}(u - z)^2$, balanced by a regularization parameter α . The resulting Euler-Lagrange equation for this problem is:

$$-\alpha \Delta u + u = z$$

with homogeneous Neumann boundary condition $\frac{\partial u}{\partial n} = 0$, which can be solved efficiently using, for example, a multigrid method (see [8]). The problem with this approach is that although smooth regions in the image are recovered well, edges present in the original image are blurred in the reconstruction.

To overcome the poor edge-capturing properties of H^1 regularization, Rudin, Osher and Fatemi (ROF) [37] proposed replacing the $\int |\nabla u|^2$ regularization term with the so-called total-variation (TV) semi-norm $\int |\nabla u|$ which will allow piecewise smooth images. The resulting minimization problem is:

$$\min_u J_{TV}(u), \quad J_{TV}(u) = \int_{\Omega} \alpha \sqrt{|\nabla u|^2 + \beta} + \frac{1}{2}(u - z)^2 dx dy \quad (1)$$

where β is a small perturbing parameter that prevents degeneracy of the Euler-Lagrange equation when $|\nabla u| = 0$. The Euler-Lagrange equation for this problem is

$$-\alpha \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) + u = z \quad (2)$$

with homogeneous Neumann boundary condition $\frac{\partial u}{\partial n} = 0$. Unlike in the H^1 case this equation is highly nonlinear and the fast solution of this equation has been an active area of research over the last decade or so.

The simplest approach is the artificial time marching (or gradient descent) method used by ROF [37]. In this method the parabolic equation

$$u_t = \alpha \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) + (z - u) \quad (3)$$

is solved to steady state using an explicit time marching (forward Euler) scheme on the discrete equation. A steepest descent type method with a line search on the discretization of $J_{TV}(u)$ can be viewed as an explicit time marching scheme with variable time step. The problem with the time marching approach is that due to stability restrictions the time step must be taken to

be very small, resulting in very slow convergence. Marquina and Osher [34] reduce the stability constraints on the time step by multiplying (3) by $|\nabla u|$.

Vogel and Oman [43] proposed a ‘lagged diffusivity’ fixed-point iterative method (see also [42]) for solving (2) in which the following linear equation is solved on each step

$$u^{k+1} - \alpha \nabla \cdot \left(\frac{\nabla u^{k+1}}{\sqrt{|\nabla u^k|^2 + \beta}} \right) = z$$

to update u . This method is equivalent to a semi-implicit time marching method with infinite time step and is globally convergent with linear convergence. The linear system to be solved on each step is symmetric positive definite and several different methods have been used in the literature to solve it, these include preconditioned conjugate gradient (pcg) with incomplete Cholesky preconditioner [14], geometric multigrid [41] (either on its own or as a preconditioner for preconditioned conjugate gradient) and Algebraic multigrid [17], which is more robust with respect to small values of β than geometric multigrid. In practice accurate solution of the linear equation is not necessary, and reducing the linear residual by a factor of 10 is usually enough to give a method which is optimal in terms of time taken.

Chan, Zhou and Chan [16] recognized that Newton’s Method has a small domain of convergence for this problem particularly with respect to small values of β and proposed a continuation procedure on β . To overcome this in a more fundamental way Chan, Golub and Mulet [14] replace the original problem with an equivalent (u, w) system

$$\begin{aligned} -\alpha \nabla \cdot w + u - z &= 0 \\ w \sqrt{|\nabla u|^2 + \beta} - \nabla u &= 0, \quad \|w\|_\infty \leq 1 \end{aligned}$$

by introducing a new variable $w = \nabla u / \sqrt{|\nabla u|^2 + \beta}$. Alternatively this system can be seen as the conditions needed to bridge the duality gap between the primal and dual problems. The new system is better behaved with respect to Newton’s method due to its quasi-linearity, and the cost of each step is only slightly more than for the primal problem (2). The linear solve on each step is done using a preconditioned conjugate gradient method with incomplete Cholesky preconditioner. The method appears globally convergent with quadratic convergence. The primal-dual method incorporates the primal and the dual variable, other authors have worked directly with the dual formulation of the TV problem see Carter [5] and more recently Chambolle [6], this avoids the use of the β parameter. Incidentally, one may eliminate u in (1) to derive the dual method [6].

In [38] we proposed using a nonlinear multigrid (FAS) method (see, for example, [39, 23]) to solve (2). Our method which used a smoother based on the fixed point method but using just 3 steps of Gauss-Seidel on the linear system on each step performed well in comparison with the fixed point and

primal-dual methods provided the parameter β was not too small. Nonlinear multigrid methods for total variation denoising have also been studied by Frohn-Schauf, Henn and Witsch in [22]. Chan and Chen [9, 10] and Chen and Tai [19] have multilevel methods working directly with the minimization problem (not the Euler-Lagrange equation). Finally another approach to solving (1) is the active set methods of Karkkainen and Majava [27] and Ito and Kunisch [24].

Although TV regularization is very good at recovering edges and blocky images it does suffer from the ‘staircasing effect’ in which smooth regions (ramps) in the original image are recovered as piecewise smooth regions (staircases) in the reconstruction. In the literature there have been many attempts to devise image denoising methods which reduce the staircasing effect seen in images denoised using TV regularization with no one approach gaining universal appeal. In most cases the emphasis is on designing new suitable regularization functionals which reduce staircasing as well as recover edges via retaining some form of the TV regularization. However no particular attention has been paid to the fast efficient solution of the resulting equations; in fact, some of these new models cannot be solved efficiently. This paper is thus motivated to study how effectively a class of staircasing reduction models can be solved by three selected iterative methods.

The underlying Euler-Lagrange equation, to be studied here, is of the form

$$-\alpha \nabla \cdot \left(D(\sqrt{|\nabla u|^2} + \beta) \nabla u \right) + u = z \quad (4)$$

with various choices of $D(t)$, where $D(t) = 1/t$ reproduces the standard ROF model. Our three selected iterative methods are: (i). the fixed point method; (ii) the time-marching methods; (iii) the nonlinear multigrid method as proposed in [38]. Therefore, our objective is to find the most reliable model by two criteria: quality of restoration results and efficiency of a fast iterative solution. It turns out that a modified model out of the tested models fits our criteria.

The rest of the paper is organized as follows. In Section 2 we review the various staircase reducing models that have been proposed in the literature including the class of models that we study in this paper. In Section 3 we discuss numerical solution of 4 particular PDE models in this class: discretization and iterative algorithms as well as the implementation of the iterative methods for each of the models and some numerical results. Section 4 focuses on the model which out of those tested we feel is best both in terms of reconstructed image quality and robustness of solvers and considers an effective modification to the best approach found from Section 3. Finally in Section 5 some conclusions are drawn.

2 An Overview of Staircasing Reduction Models

In this section we review various models to reduce the staircasing effect that have been proposed in the literature. In the next section we shall focus on the first class of the models.

2.1 Combining TV and H^1

A popular approach to reducing staircasing is to try and combine the ability of TV denoising to preserve edges with the ability of H^1 denoising to preserve smooth regions. In this paper we consider 4 such approaches for which we will attempt to use nonlinear multigrid to solve the resulting PDEs, they all involve a minimisation problem of the form

$$\min_u \int_{\Omega} \alpha \Phi(|\nabla u|) + \frac{1}{2}(u - z)^2 dx dy$$

which has the Euler-Lagrange equation

$$-\alpha \nabla \cdot \left(\Phi'(\sqrt{|\nabla u|^2 + \beta}) \frac{\nabla u}{|\nabla u|} \right) + (u - z) = 0$$

where a small parameter $\beta > 0$ is added (as in the TV case with $\Phi(g) = g$) to avoid $|\nabla u| = 0$.

Model 1

Noting that the H^1 case corresponds to $\Phi(g) = g^2$, one can propose as in [2, 29]:

$$\Phi(|\nabla u|) = \frac{1}{p} |\nabla u|^p \quad (5)$$

we then have

$$\Phi'(|\nabla u|) = |\nabla u|^{p-1} \text{ and } \frac{\Phi'(|\nabla u|)}{|\nabla u|} = \frac{1}{|\nabla u|^{2-p}}$$

where $1 < p < 2$ and in order to recover edges reasonably well p should be close to 1, say 1.1.

Model 2

A more sophisticated approach would be to choose p in some way adapting to the behavior of $|\nabla u|$. To this end, Blomgren [1] and Blomgren, Chan, Mulet [2] proposed the following general choice

$$\Phi(|\nabla u|) = |\nabla u|^{p(|\nabla u|)}, \quad \text{with } \lim_{g \rightarrow 0} p(g) = 2 \text{ and } \lim_{g \rightarrow \infty} p(g) = 1 \quad (6)$$

which results in a non-convex minimisation problem, where $p(g)$ is a monotonically decreasing function i.e TV-like regularization ($p = 1$) is used at edges, H^1 -like regularization ($p = 2$) is used in flat regions and in between $p \in (1, 2)$. We have

$$\Phi'(|\nabla u|) = p(|\nabla u|)|\nabla u|^{p(|\nabla u|)-1} + p'(|\nabla u|)|\nabla u|^{p(|\nabla u|)} \log(|\nabla u|).$$

Blomgren [1] suggests the following particular choice for p :

$$p(g) = \begin{cases} 2, & g = 0 \\ ag^3 + bg^2 + cg + d, & 0 < g < sg_{max} \\ 1, & g \geq sg_{max} \end{cases} \quad (7)$$

where the third order polynomial is chosen so that $p'(0) = 0$ and $p'(sg_{max}) = 0$, g_{max} is the maximum realizable gradient and $0 < s \leq 1$. Resolving the conditions on p gives $a = \frac{2}{(sg_{max})^3}$, $b = \frac{-3}{(sg_{max})^2}$, $c = 0$ and $d = 2$. If we assume that our image is a square $n \times n$ image with values in the range $[0, 255]$, then $g_{max} = 255\sqrt{2}(1/h)$ where h is the grid spacing (see later). We note here that in a later paper, Chan, Esedoglu, Park and Yip [13] suggested taking p to be a monotonically decreasing function from 2 to 0 e.g., $p(g) = \frac{2}{1+2g}$; here we focus on the case where p takes values between 2 and 1.

Model 3

A simplified alternative to (6) would be to replace $p = p(|\nabla u|)$ by $p = p(|\nabla u^*|)$ for some ‘known’ quantity u^* approximating u (and thus ‘less’ nonlinear), i.e., take

$$\Phi(|\nabla u|) = \frac{1}{p(|\nabla u^*|)} |\nabla u|^{p(|\nabla u^*|)} \quad (8)$$

$$\Phi'(|\nabla u|) = |\nabla u|^{p(|\nabla u^*|)-1} \text{ and } \frac{\Phi'(|\nabla u|)}{|\nabla u|} = \frac{1}{|\nabla u|^{2-p(|\nabla u^*|)}}.$$

This choice ensures that the new minimization problem is convex. Blomgren [1] suggests $u^* = G * z$ where G is a Gaussian used to smooth the noisy image z . More recently this approach was used by Karkkainen and Majava [28] who suggest $u^* = u_{TV}$ and

$$p(|\nabla u_{TV}|) = \begin{cases} 2 & |\nabla u_{TV}| < g_1 \\ 1.5 & |\nabla u_{TV}| = g_1 \\ p_1(|\nabla u_{TV}|) & g_1 < |\nabla u_{TV}| < g_2 \\ 1 & |\nabla u_{TV}| \geq g_2 \end{cases} \quad (9)$$

where $p_1(g)$ is a second order polynomial satisfying $p_1(g_1) = 1.5$, $p_1(g_2) = 1$ and $p'(g_2) = 0$. The idea here is that a value of 1.5 is enough to recover smooth regions effectively with larger values possibly oversmoothing the image. In

order that a nonlinear CG solver can be implemented effectively p takes values 2 for $|\nabla u_{TV}| < g_1$ where g_1 is small, p then jumps to a value of 1.5 and then decreases smoothly as $|\nabla u_{TV}|$ increases until $|\nabla u_{TV}| = g_2$, g_2 being small enough so that $p = 1$ at all edges in the image. The values of g_1 and g_2 are chosen using a histogram of $|\nabla u_{TV}|$ values.

Another similar, but slightly different, approach is used by Chen, Levine and Rao [20]:

$$\Phi(|\nabla u|) = \begin{cases} \frac{1}{p(|\nabla u^*|)} |\nabla u|^{p(|\nabla u^*|)} & |\nabla u| \leq \epsilon \\ |\nabla u| - \frac{\epsilon p(|\nabla u^*|) - \epsilon^{p(|\nabla u^*|)}}{p(|\nabla u^*|)} & |\nabla u| > \epsilon \end{cases}$$

$$p(|\nabla u^*|) = 1 + \frac{1}{1 + k|\nabla u^*|^2}$$

where $u^* = G * z$. The difference here is that the threshold for a switch to pure TV regularization is based on the gradient of u rather than u^* (so implicitly ‘nonlinear’). The function p is a monotonically decreasing rational function which is 2 at $|\nabla u^*| = 0$ and tends to 1 as $|\nabla u^*|$ tends to infinity. Some theoretical study of this model was conducted in [20].

Model 4

The approach proposed in [2, 1] tries to combine TV and H^1 in a convex combination:

$$\Phi(|\nabla u|) = \pi(|\nabla u|)|\nabla u| + (1 - \pi(|\nabla u|))|\nabla u|^2 \quad (10)$$

with $\lim_{g \rightarrow 0} \pi(g) = 0$ and $\lim_{g \rightarrow \infty} \pi(g) = 1$. In this case

$$\Phi'(|\nabla u|) = \pi'(|\nabla u|)(|\nabla u| - |\nabla u|^2) + \pi(|\nabla u|)(1 - 2|\nabla u|) + 2|\nabla u|.$$

It is suggested in [1] to take $\pi(g) = 2 - p(g)$ where p is the polynomial outlined in (7).

Finally we mention several other (less representative) methods which also compromise between TV and H^1 regularization. The first is the inf-convolution of the TV and H^1 regularization functionals proposed in [7] the resulting minimization problem is equivalent to:

$$\min_u \int_{|\nabla u| \geq \epsilon} |\nabla u| dx dy + \frac{\epsilon}{2} \int_{|\nabla u| < \epsilon} |\nabla u|^2 dx dy + \frac{\lambda}{2} \int_{\Omega} (u - z)^2 dx dy.$$

Another approach proposed by Ito and Kunisch [25] is to minimize the functional

$$\int_{\Omega} \alpha \Phi(|\nabla u|) + \frac{1}{2} (u - z)^2 dx dy$$

where Φ is chosen so that it behaves like $|\nabla u|$ for both large and (in contrast to other models seen above) small values of $|\nabla u|$ and behaves like $|\nabla u|^2$ for mid range values of $|\nabla u|$.

2.2 Higher Order Models

Another popular way to reduce staircasing is to introduce in some way higher order derivatives into the regularization term. In [7] Chambolle and Lions do this by minimizing the inf-convolution of the TV norm and a second order functional

$$\min_{u_1, u_2} \int_{\Omega} |\nabla u_1| + \mu |\nabla(\nabla u_2)| + \frac{\lambda}{2} (u_1 + u_2 - z)^2 dx dy. \quad (11)$$

Here u is decomposed into a smooth function u_2 and a non-smooth function u_1 containing the discontinuities. Another way to use higher order derivatives is introduced by Chan et al [15] in which the non-convex functional

$$\int_{\Omega} \left[\alpha \sqrt{|\nabla u|^2 + \beta} + \mu \frac{(\Delta u)^2}{(\sqrt{|\nabla u|^2 + 1})^3} + \frac{1}{2} (u - z)^2 \right] dx dy$$

is minimized. Here the $(|\nabla u|^2 + 1)^{-3/2}$ term multiplying the higher order term ensures that true edges (with very large gradient) are not penalized while staircasing is reduced. Instead of combining the TV norm and second order derivatives within one regularization functional Lysaker and Tai [33] use two regularization functionals:

$$E_1(u) = \int_{\Omega} |\nabla u| + \frac{\lambda_1}{2} (u - z)^2 dx dy$$

$$E_2(v) = \int_{\Omega} (v_{xx}^2 + v_{xy}^2 + v_{yx}^2 + v_{yy}^2)^{1/2} + \frac{\lambda_2}{2} (v - z)^2 dx dy.$$

Their approach is to use an iterative procedure in which they simultaneously apply an explicit time marching method to the Euler-Lagrange equation of each functional. After each step the current iterates u^k and v^k are combined in a convex combination to give $w = \theta^k u^k + (1 - \theta^k) v^k$; u^k and v^k are then overwritten with w in preparation for the next step. Here θ^k is chosen to be 1 only at the largest jumps (edges) allowing smaller jumps due to staircasing to be suppressed by the higher order PDE. In an earlier paper the same authors with Lundervold [32] considered E_2 on its own and another functional $\int_{\Omega} |u_{xx}| + |u_{yy}| + \lambda/2 (u - z)^2 dx dy$ which was not rotationally invariant.

2.3 Other Ways to Reduce Staircasing

Marquina and Osher [34] preconditioned the right hand side of the parabolic equation (3) with $|\nabla u|$ which had a staircase reducing effect. This is because the inclusion of β only in the $|\nabla u|$ term multiplying the first term on the right hand side of (3) and the use of an upwind difference scheme for the $|\nabla u|$ multiplying the second term leads to a different numerical steady state

which is less staircased than the TV problem. In a similar vein is the algebraic scaling approach used in [26] which is equivalent to using

$$u_t = \min \left(\frac{a_{max}}{2} (|\nabla u|^2 + \beta)^{1/2}, 1 \right) \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) + \lambda(z - u)$$

where a_{max} is a parameter to be chosen. We also mention the Gauss-curvature driven diffusion approach (not related to any optimization problem) proposed in [30] which has several desirable properties including staircase reduction:

$$u_t = \nabla \cdot \left(\left| \frac{u_{xx}u_{yy} - u_{xy}^2}{(1 + u_x^2 + u_y^2)^2} \right| \nabla u \right).$$

See also [4, 35], [11], [12, 36, 40, 46] for the iterated TV model, the TV L^1 model and the texture models respectively.

3 Algorithms for the Combined TV and H^1 Models

Our aim in this paper is first to implement and compare 3 numerical algorithms for solving the above listed 4 combined TV and H^1 models, and then to propose a modified staircasing reduction model which can be efficiently solvable. The selected algorithms are: (i) explicit time marching methods; (ii) fixed point type methods; (iii) the nonlinear multigrid method [38]. We now outline our discretization scheme, introduce the iterative methods and give details of implementation and numerical results.

Remark 1. As mentioned earlier less focus has been given to the efficient solution of the models of the previous section than their effectiveness in reducing staircasing. In [2] a fixed point type method is proposed to solve model 2 and model 4 but no numerical results are given. In [28] a nonlinear conjugate gradient method is used to solve model 3 with the particular choice of p outlined above. In the case of model 1 and model 3, the choice of D in (4) is similar to the TV case with the added advantage in model 3 that when $|\nabla u|$ is small $p(|\nabla u|)$ should be close to 2, preventing jumps in the diffusion coefficient as large as in the TV case. For models 2 and 4 the Euler-Lagrange equation is more nonlinear than in the TV case. We also note that many iterative methods can benefit from using the separate acceleration technique of [44].

Discretization

Below we outline the discretization scheme used. Given that the image data will be given in the form of $n \times m$ pixel values, each representing average light intensity over a small rectangular portion of the image, we use a cell-centered discretization of our domain and a cell-centered finite difference scheme to

discretize (2). From now on we assume that $\Omega = [0, n] \times [0, m]$. We discretize the domain Ω into Ω^h with $n \times m$ rectangular cells of size $h \times k$ where $h = k = 1$, with grid points placed at the center of the cells so grid point (i, j) is located at

$$(x_i, y_j) = ((2i - 1)h/2, (2j - 1)k/2).$$

Denoting the discrete version of equation (4) by $N_h(u_h) = z_h$, we have:

$$(N_h(u_h))_{i,j} = u_{i,j} - \alpha_h [\delta_x^- (D_{ij}(g_{ij})\delta_x^+ u_{i,j}) + \gamma\delta_y^- (D_{ij}(g_{ij})\gamma\delta_y^+ u_{i,j})] = z_{ij} \quad (12)$$

where u_h and z_h are grid functions on Ω^h ,

$$g_{i,j} = \frac{1}{h} \sqrt{(\delta_x^+ u_{i,j})^2 + (\gamma\delta_y^+ u_{i,j})^2 + \beta_h}$$

$$D_{ij}(g_{ij}) = \begin{cases} \frac{1}{h}(g_{ij}^{-(2-p)}) & \text{Model 1} \\ \frac{1}{h}(p(g_{ij})g_{ij}^{p(g_{ij})-1} + p'(g_{ij})g_{ij}^{p(g_{ij})} \log(g_{ij}))g_{ij}^{-1} & \text{Model 2} \\ \frac{1}{h}(g_{ij}^{-(2-p_{i,j})}) & \text{Model 3} \\ \frac{(\pi'(g_{ij})(g_{ij}-g_{ij}^2)+\pi(g_{ij})(1-2g_{ij})+2g_{ij})}{hg_{ij}} & \text{Model 4} \end{cases} \quad (13)$$

$$\alpha_h = \alpha/h, \beta_h = h^2\beta \text{ and } \gamma = h/k = 1$$

and

$$\delta_x^\pm u_{i,j} = \pm (u_{i\pm 1,j} - u_{i,j}) \quad \delta_y^\pm u_{i,j} = \pm (u_{i,j\pm 1} - u_{i,j}).$$

Note that D is actually only dependant on (i, j) in the case of model 3. We also have boundary condition:

$$u_{i,0} = u_{i,1}, u_{i,m+1} = u_{i,m}, u_{0,j} = u_{1,j}, u_{n+1,j} = u_{n,j}. \quad (14)$$

Remark 2. Unlike in the TV case where the choice of Ω is not important provided α_h and β_h are chosen to be the same, whatever the value of h , there is not in all cases here a straightforward relationship (the exception is model 1) between the case $\Omega = [0, n] \times [0, m]$ i.e $(h, k) = (1, 1)$ and the case $\Omega = [0, 1] \times [0, 1]$ i.e $(h, k) = (1/n, 1/m)$. We have chosen the former to be consistent with the majority of papers.

We now introduce the 3 algorithms to be used.

Algorithm 1 (Time Marching)

Choose initial guess u_h^0

Set $k = 0$.

While $\|vec(z_h - N_h(u_h^k))\|_2 > tol$

$$u_h^{k+1} = u_h^k + \Delta t [z_h - N_h(u_h^k)]$$

$$k = k + 1$$

end

The time step Δt is determined by experiment as the largest value which gives stability of the algorithm. Here vec denotes the stacking of a grid function into a vector and tol is typically $10^{-4} \|\text{vec}(z_h - N_h(z_h))\|_2$, where $\|\cdot\|_2$ is the Euclidean norm.

Algorithm 2 (Fixed Point Method)

Choose initial guess u_h^0 and Set $k = 0$.

While $\|\text{vec}(z_h - N_h(u_h^k))\|_2 > \text{tol}$

Set u_h^{k+1} to be the result of applying some iterative method to:

$$L_h(u_h^k)w_h = z_h$$

$$k = k + 1$$

end

The linear operator $L_h(u_h^k)$ on step $k + 1$ is given by the stencil:

$$\begin{bmatrix} 0 & -\alpha\lambda D_{ij}(g_{ij}^k) & 0 \\ -\alpha D_{i-1,j}(g_{i-1,j}^k) & 1 + \alpha\Pi_{ij} & -\alpha D_{ij}(g_{ij}^k) \\ 0 & -\alpha\lambda D_{i,j-1}(g_{i,j-1}^k) & 0 \end{bmatrix}$$

where $\Pi_{ij} = (1+\lambda)D_{ij}(g_{ij}^k) + D_{i-1,j}(g_{i-1,j}^k) + \lambda D_{i,j-1}(g_{i,j-1}^k)$. The linear solver used in most cases is a geometric multigrid method with red-black Gauss-Seidel pre-correction and black-red Gauss-Seidel post correction as smoother (c.f. [41, 43]). We only require a relatively small decrease in the linear residual (typically a halving) as this seems to give the best results in terms of overall cpu time. We may also require the use of methods such as preconditioned conjugate gradient and minimum residual, we stack the grid functions along rows of pixels into vectors $\mathbf{u}_h^k = (u_{1,1}^k, u_{2,1}^k, \dots, u_{n,1}^k, u_{1,2}^k, \dots, u_{n,m}^k)^T$ and $\mathbf{z}_h$, the resulting system is of the form $A(\mathbf{u}_h^k)\mathbf{w}_h = \mathbf{z}_h$ where A is symmetric.

Nonlinear Multigrid

Multigrid methods based on the recursive application of smoothing relaxation and coarse grid correction are efficient solvers for a wide range of linear and nonlinear elliptic partial differential equations. Below we give a brief introduction to the full approximation scheme (FAS) nonlinear multigrid scheme and review the smoother used in [38] for the TV problem before giving the algorithm for a similar scheme to be used in this paper. For a more comprehensive treatment of multigrid see for example [3, 18, 39, 45] and references therein.

Denote by $N_h u_h = z_h$ the nonlinear system (12) and by Ω^{2h} the $n/2 \times m/2$ cell-centered grid which results from standard coarsening of Ω^h i.e the cell-centered grid with grid spacing $(2h, 2k)$. If v_h is an approximation to the solution u_h define the error in v_h by $e_h = u_h - v_h$ and the residual by $r_h = z_h - N_h v_h$ recall also that these quantities are related by the nonlinear residual equation:

$$N_h(v_h + e_h) - N_h v_h = r_h.$$

If e_h is ‘smooth’ it can be well approximated on Ω^{2h} . To describe a multigrid cycle, we define the transfer and smoothing operators.

The **Restriction** operator is

$$I_h^{2h} v_h = v_{2h}$$

where

$$(v_{2h})_{i,j} = \frac{1}{4}[(v_h)_{2i-1,2j-1} + (v_h)_{2i-1,2j} + (v_h)_{2i,2j-1} + (v_h)_{2i,2j}]$$

$$1 \leq i \leq n/2, 1 \leq j \leq m/2.$$

The **Interpolation** operator is defined by

$$I_{2h}^h v_{2h} = v_h$$

where

$$(v_h)_{2i,2j} = \frac{1}{16}[9(v_{2h})_{i,j} + 3[(v_{2h})_{i+1,j} + (v_{2h})_{i,j+1}] + (v_{2h})_{i+1,j+1}]$$

$$\frac{1}{16}[9(v_{2h})_{i,j} + 3[(v_{2h})_{i-1,j} + (v_{2h})_{i,j+1}] + (v_{2h})_{i-1,j+1}]$$

$$\frac{1}{16}[9(v_{2h})_{i,j} + 3[(v_{2h})_{i+1,j} + (v_{2h})_{i,j-1}] + (v_{2h})_{i+1,j-1}]$$

$$\frac{1}{16}[9(v_{2h})_{i,j} + 3[(v_{2h})_{i-1,j} + (v_{2h})_{i,j-1}] + (v_{2h})_{i-1,j-1}]$$

$$1 \leq i \leq n/2, 1 \leq j \leq m/2.$$

Local smoothers. At grid point (i, j) the Euler-Lagrange equation is

$$u_{i,j} - \alpha_h (D_{i,j}(g_{i,j})(u_{i+1,j} - u_{i,j}) - D_{i-1,j}(g_{i-1,j})(u_{i,j} - u_{i-1,j})$$

$$+ \gamma^2 [D_{i,j}(g_{i,j})(u_{i,j+1} - u_{i,j}) - D_{i,j-1}(g_{i,j-1})(u_{i,j} - u_{i,j-1})]) = z_{i,j} \quad (15)$$

where g_{ij} depends on $u_{i+1,j}$, $u_{i,j+1}$ and $u_{i,j}$. If we freeze all non (i, j) terms at the current approximation then we have a nonlinear equation in one variable to solve in order to update the approximation at (i, j) , which can be done using a step of Newton’s method. This type of local nonlinear relaxation scheme is known as Gauss-Seidel Newton. In our investigations into the TV problem we found that this type of method only converged with heavy under-relaxation and was not useful as a smoother for the nonlinear multigrid method. An alternative approach would be to freeze also the g terms in (15) rather than just the $u_{i,j}$ terms in g . In this case we have a linear equation in one variable to solve at each grid point. This type of approach is more stable than Gauss-Seidel Newton and can be speeded up in the TV case by the application of nonlinear multigrid, however we found in [38] that a better option is a smoother in which the Euler-Lagrange equation is linearized globally as in the fixed point method before a few (3 seems to be optimal) steps of linear Gauss-Seidel relaxation are applied to the linear system i.e $D_{ij}(g_{ij})$ is evaluated for all (i, j) at the beginning of the smoothing step using the value of the current iterate before linear Gauss-Seidel is used to update. We call this smoother FPGS. For clarity the algorithm for one step of the FPGS smoother is given below

```


$$v_h \leftarrow FPGS(v_h, N_h, z_h)$$

for  $i = 1 : n$ 
  for  $j = 1 : m$ 
    Evaluate  $g_{i,j} = ((\delta_x^+ v_{i,j})^2 + (\gamma \delta_y^+ v_{i,j})^2 + \beta_h)^{-1/2}$ 
    and  $D_{i,j}(g_{i,j})$  according to  $N_h$  using (13).
  end
end
Perform Gauss-Seidel steps on linear system (start from  $w = v_h$ )
for  $iter = 1 : it$ 
  for  $j = 1 : m$ 
    for  $i = 1 : n$ 
       $\bar{w} \leftarrow w, T_0 = D(g_{i,j})_{i,j}, T_1 = D(g_{i-1,j})_{i-1,j}, T_2 = D(g_{i,j-1})_{i,j-1},$ 

$$w_{i,j} \leftarrow \frac{z_{i,j} + \alpha_h(T_0(\bar{w}_{i+1,j} + \gamma^2 \bar{w}_{i,j+1}) + T_1 \bar{w}_{i-1,j} + \gamma^2 T_2 \bar{w}_{i,j-1})}{1 + \alpha_h((1 + \gamma^2)D(g_{i,j})_{i,j} + T_1 + \gamma^2 T_2)}$$

      or an appropriate modification if  $(i, j)$  is a boundary point.
    end
  end
end
 $v_h \leftarrow w_h$ 

```

We take $it = 3$ unless otherwise stated.

Any iterative method which smooths the error on the fine grid i.e damps high frequency Fourier components of the error while not necessarily reducing its size greatly can be improved by the use of coarse grid correction, in which a coarse grid analogue of the residual equation is solved (solution on the coarse grid being less expensive than on the fine grid) to obtain a coarse grid approximation of the error, which is then transferred back to the fine grid to correct the approximation v_h .

The Nonlinear Multigrid Method

We are ready to state the algorithm for the FAS multigrid method with FPGS smoother that we use in this paper. The method is a V-cycle method, which means that just one recursive call to the algorithm is made on each level to approximately solve the coarse grid problem, we have found that using the more expensive W-cycle (performing two cycles to solve the coarse grid problem on each level) does not give a significant improvement in convergence and therefore is not pursued.

Algorithm 3 (Nonlinear Multigrid Method)

```

Set  $v_h$  to be some initial guess.
While  $\|vec(z_h - N_h(v_h))\|_2 > tol$ 
   $v_h \leftarrow NLMG^h(v_h, N_h, z_h, \nu_1, \nu_2)$ 
end

```

where $NLMG^h$ is defined recursively as $v^h \leftarrow NLMG^h(v_h, N_h, z_h, \nu_1, \nu_2)$ as follows:

1. If $\Omega^h = \text{coarsest grid}$, solve $N_h u_h = z_h$ using Fixed Point Method and stop.
- Else For $l = 1, \dots, \nu_1$ $v_h \leftarrow FPGS(v_h, N_h, z_h)$
2. $v_{2h} = I_h^{2h} v_h$, $\bar{v}_{2h} = v_{2h}$, $z_{2h} = I_h^{2h}(z_h - N_h v_h) + N_{2h} v_{2h}$
3. $v_{2h} \leftarrow NLMG^{2h}(v_{2h}, N_{2h}, z_{2h}, \nu_1, \nu_2)$
4. $v_h \leftarrow v_h + I_{2h}^h(v_{2h} - \bar{v}_{2h})$
5. For $l = 1, \dots, \nu_2$ $v_h \leftarrow FPGS(v_h, N_h, z_h)$

Here $v_h \leftarrow FPGS(v_h, N_h, z_h)$ denotes the updating of v_h via one step of the FPGS smoother. N_{2h} is the coarse grid analogue of N_h which results from standard coarsening i.e the nonlinear operator which results from discretizing the Euler-Lagrange equation using a cell-centered grid with grid spacing $(2h, 2k)$. The number of pre and post-correction smoothing steps (ν_1 and ν_2) we use depends on the model under consideration, details will be given below. We use standard cell-centered interpolation and restriction operators outlined earlier, and take the coarsest grid as 4×4 .

Numerical Results

Now we present some numerical results and give details of some of the issues regarding our implementation of iterative methods for each of the four models. It should be remarked that although Algorithms 1-2 have been used for solving some of these equations it is up to now unclear whether Algorithm 3 would work for the models considered.

Tests are run on the test hump image seen in Figure 1, which has both smooth regions, high intensity edges and low intensity edges and the more realistic Lenna image shown in Figure 2. In each case we have tried to choose parameters which give the optimal reconstruction, focusing on the need to reduce staircasing. What the optimal reconstruction is, is somewhat subjective, as a guide we have used mesh and image plots as well as Peak signal to noise ratio (PSNR) defined by

$$PSNR = 20 \log_{10} \left(\frac{255}{RMSE(u, u^0)} \right), \quad RMSE(u, u^0) = \sqrt{\frac{\sum_{(i,j)} (u_{i,j} - u_{i,j}^0)^2}{nm}}$$

where u is the reconstructed image and u^0 is the true image.

The PSNR does not always give a clear guide as to whether one image is less staircased than another as can be seen in the hypothetical 1D example in Figure 3, so we also take into account the value of $PSNR_{grad}$ which we define as $1/2(PSNR(u_x, u_x^0) + PSNR(u_y, u_y^0))$ this should measure how well the derivatives of the reconstruction match those of the true image. All methods were implemented in MATLAB on a Sun Fire 880.

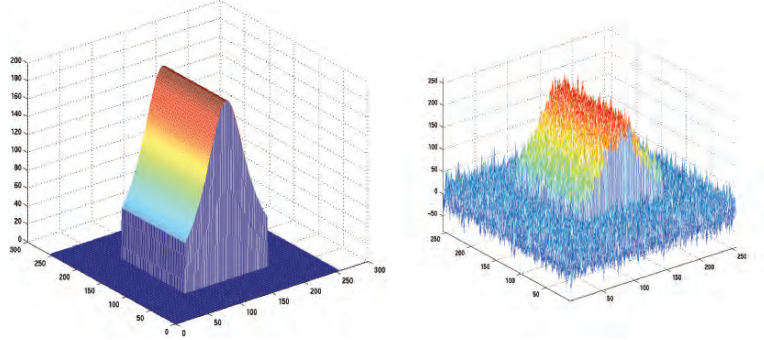


Fig. 1. Mesh plots of true (left) and noisy (right) Hump image.

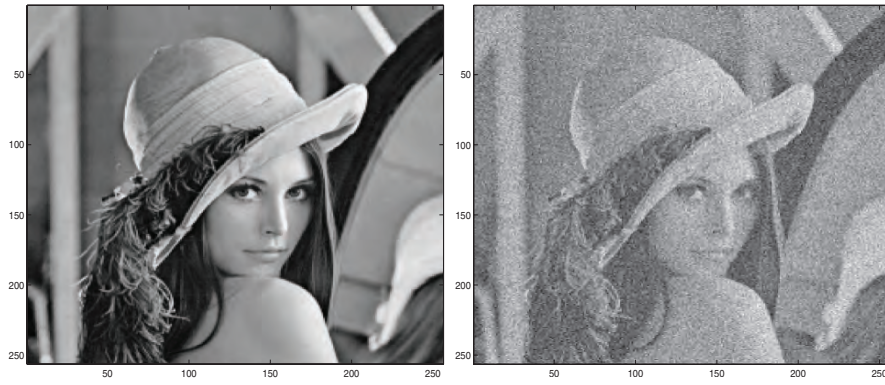


Fig. 2. True (left) and noisy (right) Lenna image.

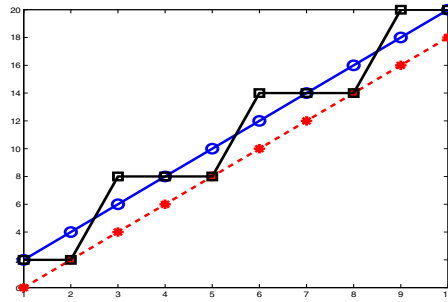


Fig. 3. A simple 1D example of a staircased reconstruction (squares) which will have a higher PSNR than the smooth reconstruction (stars), the smooth reconstruction in this case has exactly the same gradient as the true solution (circles).

In Figure 4, we present some plots showing the results of applying each of the four models to the test hump image, we also show the results of applying TV and H^1 regularization. We remark that it is not our intention in this paper to carry out a detailed comparison of the various staircase reducing methods in terms of the quality of the reconstructed images, however we make a few general comments. To some extent all the models can recover better the smooth regions of the image than the original TV model (1) but in our experience models 2 and 3 seem to give better overall results than model 1 (as would be expected) and model 4 in which there is some over-smoothing of the edges (particularly the low intensity edges), as noted in [1]. With models 2 and 3 for the test image shown we have been able (with suitable choices of parameters) to reduce the staircasing present in the TV reconstructed image while still recovering well the high and low intensity edges in the image.

Model 1

For this model we consider three choices of p , $p = 1.1$, $p = 1.5$ and $p = 1.9$ mainly to highlight the effect the value of p has on the convergence of the various methods (the latter two choices will of course over-smooth the edges). A suitable value of α_h to remove the noise is chosen for each value, the larger p is the smaller α_h needs to be. The effect that the parameter β_h has on convergence is also studied.

In Table 1 we show results (number of steps required for convergence and cpu time in seconds) for the Fixed Point method (FP), Nonlinear multigrid method (NLMG) and the explicit time marching method (TM) run on model 1 for the hump image with 3 different values of p , 1.1, 1.5 and 1.9 the corresponding values of α_h are 52, 24 and 15. Also shown are results for the smoother (FPGS) run on its own and results for various values of β_h . In all cases the initial guess is taken to be the noisy image z and the stopping criterion is a reduction in the residual by a factor of 10^{-4} . As linear solver in the fixed point method, a linear multigrid method with 2 pre and 2 post correction smoothing steps of Gauss-Seidel relaxation is used until the linear residual has been reduced by a factor of 0.5. Shown in the table are the choices of ν_1 and ν_2 which give the optimal nonlinear multigrid method for each case, also shown is the value of the time step in the time marching method.

We observe that the closer p is to 2 the easier the problem is to solve, less steps are required for each of the methods and less smoothing steps are required in the nonlinear multigrid method. We see that for $p = 1.9$ the convergence of the various methods is seemingly invariant to the value of β_h . For $p = 1.5$ decreasing the value of β_h has only a small effect on the FP method and the FPGS smoother and no effect on the nonlinear multigrid method. In the case that $p = 1.1$ the value of β_h has a significant effect on convergence. We see that as β_h is decreased from 10^{-2} to 10^{-4} the cost of the fixed point method increases by 3 times. The cost of the nonlinear multigrid method doubles and more pre and post correction steps are needed to ensure

Table 1. Comparison of the 3 main Algorithms for Model 1 with various p and β .

p	β_h	FP steps	cpu	TM Δt	steps	cpu
1.1	10^{-2}	43	73	5×10^{-4}	9502	2540
	10^{-4}	73	216			
1.5	10^{-2}	14	19	1×10^{-3}	4054	536
	10^{-4}	16	23	1×10^{-3}	4053	536
	10^{-10}	16	23	5×10^{-4}	8150	1131
1.9	10^{-2}	6	8.8	1×10^{-2}	303	56
	10^{-10}	6	8.8	1×10^{-2}	303	56

p	β_h	NLMG ν_1/ν_2	steps	cpu	FPGS steps	cpu
1.1	10^{-2}	5/5	4	34	748	680
	10^{-4}	10/10	4	66	4389	4036
1.5	10^{-2}	1/1	6	13	78	61
	10^{-4}	1/1	6	13	94	74
	10^{-10}	1/1	6	13	119	93
1.9	10^{-2}	1/1	3	6.9	29	23.9
	10^{-10}	1/1	3	6.9	29	23.9

convergence. We found that the time marching method cannot converge in a reasonable number of steps. If β_h is reduced to 10^{-10} only the fixed point method converges in a reasonable number of steps (in this case a pcg linear solver with Cholesky preconditioner gives the best results). This breakdown of the nonlinear multigrid convergence for very small β_h was also observed in the TV ($p = 1$) case. Apart from this last case the nonlinear multigrid method significantly speeds up the smoother FPGS and is faster than the time marching and fixed point methods.

Model 2

For this model $p(|\nabla u|)$ is chosen to be the polynomial (7). There were several problems that occurred during the implementation of iterative solvers for this model. The first problem is that the functional is non-convex and the initial guess seems to have an effect on the quality of the final image. If we take the noisy image z as initial guess we appear to converge to a minimum which is still highly oscillatory. To achieve the reconstruction of the test image shown in Figure 4 we had to take the solution to the TV problem as initial guess, the following discussion relates to experiments run using this initial guess. The second problem is that unlike in the TV case the D_{ij} terms can take negative values, as a consequence the previous smoother FPGS is *no longer*

adequate. We proposed a modification of this smoother (to be denoted by FPGS2). Instead of updating u^{k+1} by applying 3 Gauss-Seidel steps to the linear system $L(u_h^k)w_h = z_h$ we apply 3 Gauss-Seidel steps to the *new* linear system $(\lambda + L(u_h^k))w_h = z_h + \lambda u_h^k$ (essentially we add a λu term to both sides of the Euler-Lagrange equation and lag the right hand side term). Taking λ large enough will ensure diagonal dominance of the inner linear system and hence positive definiteness, which ensures convergence of the Gauss-Seidel steps. In addition we have also used this approach when implementing the fixed point method. We tried to implement the fixed point method in its original form but had problems finding a suitable inner solver (linear multigrid did not converge and pcg was not an option) we settled on the minimum residual method but found that the outer fixed point steps stagnated, this was also the case when we used a direct solver to solve the linear system. Using the modified fixed point method, we can use linear multigrid or pcg as the inner linear solver and the outer steps also converge.

We implemented the time marching method, the modified fixed point method and the nonlinear multigrid method with FPGS2 smoother on the test hump image using a value of $s = 0.2$, $\alpha_h = 10$ and $\lambda = 7$, in this case only 2 pre and 2 post correction smoothing steps were required in the nonlinear multigrid method which converged in 9 steps and was around 1.75 times as fast as the modified fixed point method and over 5 times as fast as the time marching method. However when we tried to implement this model for the Lenna image we could not achieve a reasonable quality reconstruction, the image tended to look too blurred or be contaminated with undesirable artifacts. In addition we found that the nonlinear multigrid method is not effective in that the convergence stagnates unless a large number (10 or more) of smoothing steps is used and the total number of smoothing steps in this case is more than if the smoother were run on its own. The convergence of the modified fixed point method also seems somewhat unstable and typically the number of steps required by the modified fixed point and time marching methods is considerably larger than the case of the hump image above. We note that some of the problems with the iterative methods described above also occur in the case of the hump image for larger values of s (although these do not produce good reconstructions). More work is needed on this model before we can draw any firm conclusions.

Finally we note that the value of β_h seems to have no effect on convergence for this model and so it is taken to be very small (10^{-10}) in the implementation.

Model 3

We have implemented model 3 with the choice of $p(|\nabla u^*|)$ described by (9). We have been able to implement a working nonlinear multigrid method (with the usual FPGS smoother) as well as the fixed point and time marching methods.

For the parameters g_1 and g_2 in (9) we take $g_1 = g_{max}^*/50$ (as in [28]) and $g_2 = sg_{max}^*$ where $0 < s < 1$ and is chosen to give the best visual results, g_{max}^*

is the maximum value of $g_{i,j}^*$ over all (i,j) where the $g_{i,j}^*$ is the discretization of $|\nabla u^*|$ at grid point (i,j) , u^* in this case being the TV solution u_{TV} .

In Table 2 (left) results of running FP, NLMG and TM on model 3 for the hump test image are shown. In this case we take $s = 0.3$ and $\alpha_h = 30$, β_h in this case appears to have no effect on convergence and is taken to be 10^{-10} . We take z as the initial guess and the same stopping criteria as above is used. One pre and one post correction smoothing step is used in the nonlinear multigrid method, for the fixed point method linear multigrid is used as the linear solver again with the same stopping criteria as in model 1. The time step in the time marching method is $\Delta t = 8.0 \times 10^{-3}$.

Table 2. Comparison of Fixed Point, Time Marching and Nonlinear Multigrid for Model 3 (top) and Model 4 (bottom) on the hump image and the Lenna image.

Method	Model 3		Lenna image	
	Hump image Steps	cpu(s)	Steps	cpu(s)
FP	8	11.8	10	13.8
FPGS	33	24.3	22	17.3
NLMG	4	8.4	5	10.5
TM	213	27.9	169	24.8

Method	Model 4		Lenna image	
	Hump image Steps	cpu(s)	Steps	cpu(s)
FP	16	17.9	22	24.7
FPGS	140	31.3	78	17.5
NLMG	6	8.0	8	10.3
TM	378	34.2	245	21.8

We observe that the nonlinear multigrid method reduces the cost of the smoother alone by approximately 65%. Nonlinear multigrid is around 1.4 times faster than the fixed point method and around 3.3 times as fast as the time marching method.

In our second test, we compare the performance of fixed point, time marching and nonlinear multigrid on the more realistic Lenna image. In this case we take $s = 0.9$ and $\alpha_h = 11$. The implementation is as above, except that the time step $\Delta t = 2.2 \times 10^{-2}$ is used in the time marching method. The usual initial guess and stopping criteria are used, results are given in Table 2 (left). In this case the speed up in the smoother achieved by the nonlinear multigrid method is around 40%, the nonlinear multigrid method is around 1.3 times as fast as the fixed point and around 2.4 times faster than the time marching method.

Model 4

We consider (10) only for the case

$$\pi(x) = \frac{\epsilon x}{\epsilon x + q} \quad (16)$$

In this case the functional is convex (see [2] for the conditions on π required for a convex functional). Also

$$D(x) = \frac{\Phi'(x)}{x} = \frac{(\epsilon + q)(\epsilon x + 2q)}{(\epsilon x + q)^2}$$

which is positive for nonnegative x ensuring a positive definite linear system in the fixed point method. With this choice we have successfully implemented nonlinear multigrid fixed point and time marching methods. With other choices of $\pi(x)$ e.g $2 - p(x)$ where p is the third order polynomial, we may not have a convex functional and some of same issues as in the case of Model 2 may arise. We are not aware of the choice (16) being used before but in our experience it is easier to implement iterative solvers for this case.

We have found that the choice of ϵ is more important than the choice of q in obtaining a reasonable reconstruction. With our choice of π the Euler-Lagrange equation is not degenerate for $|\nabla u| = 0$ and so we take $\beta_h = 0$.

In Table 2 (right) we show some results for the FP, NLMG and TM methods run on model 4 for the hump image, with the particular choice of π outlined above. For the parameters ϵ and q in π we take values 0.001 and 0.005 respectively, the value of α_h is 9. We have found in this case that the fastest multigrid method was achieved if we took the parameter *it* in the FPGS smoother to be 1 rather than the usual 3. The initial guess, stopping criteria and linear solver for the fixed point method are the same as in the case of model 1 and model 3. In the nonlinear multigrid method we use 2 pre and 2 post correction smoothing steps and in the time marching method we use a time step $\Delta t = 1.3 \times 10^{-2}$.

We observe that the nonlinear multigrid method reduces the cost of the smoother alone by around 75%. The nonlinear multigrid method is ≈ 2.2 times as fast as the fixed point method and ≈ 4.3 times as fast as the time marching method.

We also applied model 4 to the Lenna image, results are shown in Table 2 (right). The value of q and ϵ are as above, but $\alpha_h = 5$. The implementation is as above, except that the time step in the time marching method is $\Delta t = 2.7 \times 10^{-2}$. In this case the FPGS smoother on its own performs quite well and is actually slightly faster than the fixed point method with linear multigrid inner solver. The nonlinear multigrid method is 1.7 times faster than FPGS. The time marching method is actually quite competitive in this case at around twice the cost of the nonlinear multigrid method.

Remark 3. Although model 4 did not perform that well on the hump image with oversmoothing of some edges, we have observed for more realistic images

like the Lenna image, where the intensity of edges is more uniform, this model does not perform that badly in comparison with model 3 as can be seen from the plots in Figure 5.

To summarise we have successfully implemented the three iterative methods for both model 3 and model 4 with a specific choice of π . At the moment there are still some outstanding issues regarding both the robustness of iterative solvers and the quality of the reconstructed image for model 2 and model 4 with other possible π , we therefore favour model 3 with the nonlinear multigrid solver as a method which can achieve good quality reconstructions and can be solved simply and efficiently. In the next section we consider other possible choices of $p(|\nabla u^*|)$ and show that the nonlinear multigrid method is the most efficient solver.

4 A Modified Staircasing Reduction Model

We hope to improve on the above recommended model 3 further. To this end, we wish to simplify the specification of $p(v)$ in (9) while maintaining the smooth transition of $p(v) = 1$ to $p(v) = 1.5$. Our proposed modification is still of the general type (4)

$$-\alpha \nabla \cdot \left(\Phi(|\nabla u|, |\nabla u^*|) \nabla u \right) + u = z \quad (17)$$

where

$$\begin{aligned} \Phi(t, v) &= \frac{1}{p(v)} t^{p(v)} \\ p(v) &= 1.5 \left(1 + \frac{2v}{g_2} \right) \left(\frac{v - g_2}{g_2} \right)^2 + \left(1 - \frac{2(v - g_2)}{g_2} \right) \left(\frac{v}{g_2} \right)^2 \end{aligned} \quad (18)$$

and $p(t)$ is a cubic Hermite polynomial satisfying $p(g_1) = 1.5$, $p(g_2) = 1$ and $p'(g_1) = p'(g_2) = 0$ (here we take $g_1 = 0$). An alternative choice of $p(v)$ is a cubic Hermite polynomial satisfying $p(g_1) = 2$, $p(g_2) = 1$ and $p'(g_1) = p'(g_2) = 0$ (which has been found to perform similarly). Here $u^* = u_{TV}$ is the numerical solution from the standard ROF model, which distinguishes smooth regions and edges in an image. As with other models, the idea is again to respect large gradients (edges) and to reduce the effect of TV for small gradients (smooth regions). First of all, as illustrated in Figure 6, one can observe that such a model is simpler than the 'three piece' choice for $p(|\nabla u^*|)$ used in [28]. It only remains to test, in comparison to previous models, how effectively the new model can reduce the staircasing effect and how efficiently it can be solved by our selected iterative methods.

Restored Quality Comparison

Instead of visualizing the restored images, we compare the $PSNR$ values of the new model with Model 3 in Table 3. The same values of g_2 and α_h are

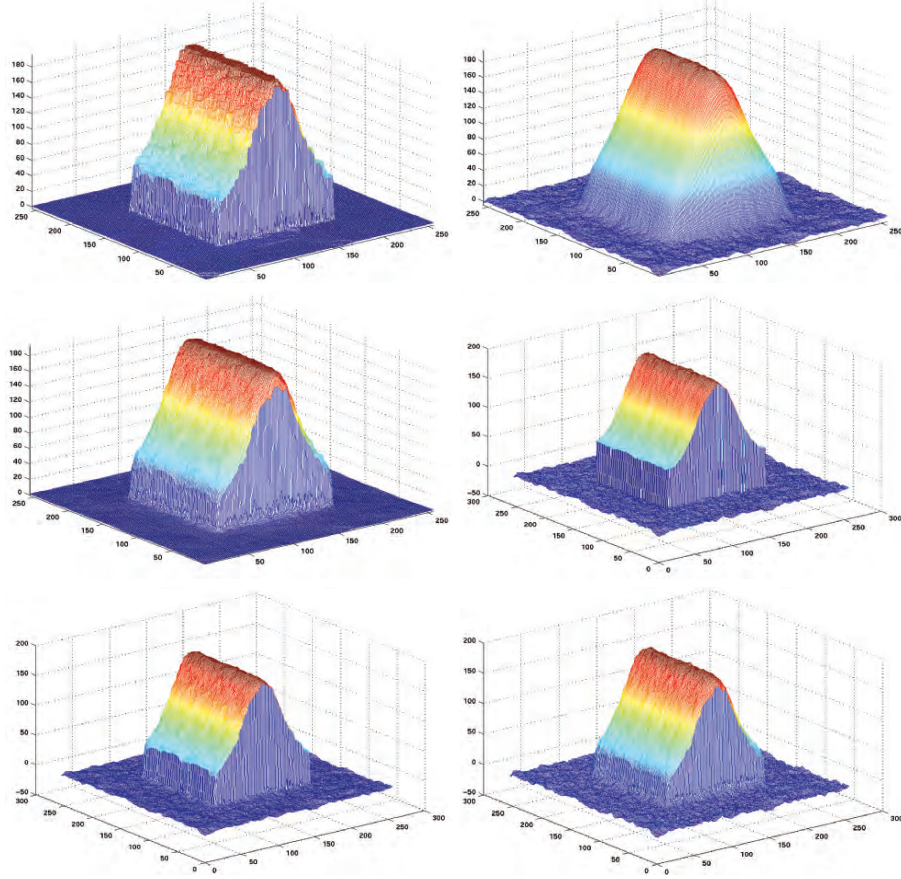


Fig. 4. From top left to bottom right, the images recovered using TV, H^1 , model 1 ($p = 1.1$), model 2, model 3 and model 4.

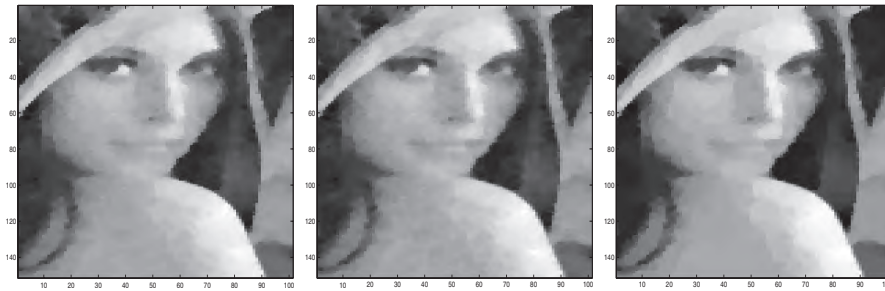


Fig. 5. Close up of Lenna Image recovered using model 3 (left) and model 4 (centre), with TV result (right) for comparison, notice the reduction in staircasing on the face and shoulder.

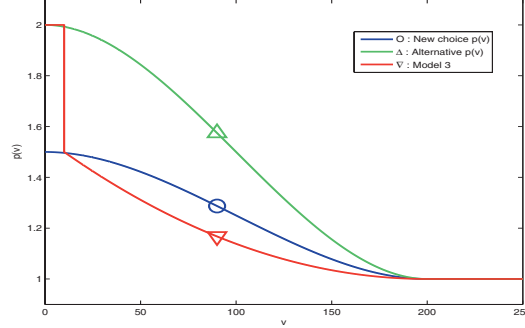


Fig. 6. Comparison of the choice of the exponent of a modified TV function.

used for both model 3 and the new model. The results presented illustrate the general trend that we have observed; for the hump image both the $PSNR$ and $PSNR_{grad}$ values are higher for the new model than for model 3 while in the case of the Lenna image we can achieve slightly higher $PSNR$ values with model 3 but the $PSNR_{grad}$ values are higher with the new model, on visual inspection Lenna's face also looks a little smoother when using the new model.

Table 3. Comparison of solution quality ($PSNR$) and speed (by MG) of Model 3 and the new model.

Image	Model 3		The new model	
	$PSNR$	$PSNR_{grad}$	$PSNR$	$PSNR_{grad}$
Hump	41.77	47.26	42.45	48.03
Lenna	28.73	28.31	28.53	28.51
(NLMG) steps		CPU	steps	CPU
Hump	4	13.1	4	14.4
Lenna	4	13.7	4	14.4

Efficiency Comparison

In terms of implementation, the new model can be solved similarly to model 3 as they are of the same model type. The cost of 4 nonlinear multigrid steps with 2 pre and 2 post correction smoothing steps is shown in Table 3. The cost of the new model per step is very slightly higher than for model 3, we

think this is because it costs slightly less in terms of cpu to evaluate $|\nabla u|^{2-p}$ when $p = 1$ or 2 than it does when $1 < p < 2$ and model 3 takes $p = 2$ when $|\nabla u^*| < g_1$.

In general the advantage of the nonlinear multigrid method over the fixed point method and, in particular, the time marching method is greater for the new model (polynomial from 1.5 to 1) than it is for model 3, the nonlinear multigrid method has been observed to be over twice as fast as the fixed point method and up to 90 times as fast as the time marching, the advantage in the case of the other polynomial (2 to 1) is similar to that observed for model 3.

Remark 4. We have considered several other choices of p and u^* which include general second and third order polynomials ranging between $2 < q < 1$ at 0 and 1 at sg_{max}^* and a rational similar to that used by [20, 21, 31] but with the threshold for TV regularization built into p , for both $u^* = u_{TV}$ and $u^* = G_\gamma * z$ where $G_\gamma = ce^{-\gamma(x^2+y^2)}$ is a Gaussian used to smooth the noisy image. Typically q should be between 1.75 and 1.5 to give the best results and 0.75 is a suitable choice for γ . For some realistic images $u^* = G_\gamma * z$ gives slightly superior results to $u^* = u_{TV}$ in our experience, although u_{TV} is better for the hump image. The nonlinear multigrid method is in all cases we have tested faster than the fixed point and time marching method.

5 Conclusion

We studied several staircasing-reducing regularization methods in this paper. Firstly we compared the efficiency of solving these models by 3 selected iterative solvers and the restored quality, and concluded that Model 3 is the most robust staircasing reduction model. Secondly we proposed a simpler model than Model 3. Numerical tests show that the new model offers better restored quality (in terms of higher PSNR values) and equally efficient solution.

Acknowledgement. The authors thank the anonymous referees for making helpful remarks and suggestions. This work is supported by an UK EPSRC DTA award and the Leverhulme Trust RF/9/RFG/2005/0482.

References

1. P. Blomgren. *Total Variation Methods for Restoration of Vector Valued Images*. PhD thesis, UCLA, 1998.
2. P. Blomgren, T. F. Chan, and P. Mulet. Extensions to total variation denoising. In *SPIE 97*, San Diego, 1997.
3. W. Briggs. *A Multigrid Tutorial*. SIAM, Philadelphia, 1987.
4. M. Burger, S. Osher, J. Xu, and G. Gilboa. Nonlinear inverse scale space methods for image restoration. Technical Report 05-34, UCLA, 2005.

5. J. Carter. *Dual Methods for Total Variation Based Image Restoration*. PhD thesis, UCLA, 2001.
6. A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imag. Vis.*, 20:89–97, 2004.
7. A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Numer. Math.*, 76:167–188, 1997.
8. R. Chan, T. F. Chan, and W. L. Wan. Multigrid for differential-convolution problems arising from image processing. In *Proceedings of the Workshop on Scientific Computing*, 1997.
9. T. F. Chan and K. Chen. On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimization, *Numer. Algorithm.*, 41:387–411, 2006.
10. T. F. Chan and K. Chen. An optimization-based multilevel algorithm for total variation image denoising. *Multiscale Model. Simul.*, 5(2):615–645, 2006.
11. T. F. Chan and S. Esedoglu. Aspects of total variation regularized l^1 function approximation. *SIAM J. Appl. Math.*, 65:1817–1837, 2005 (see also CAM04-07).
12. T. F. Chan, S. Esedoglu, and F. Park. Image decomposition combining staircase reduction and texture extraction. Technical Report 05-18, UCLA, 2005.
13. T. F. Chan, S. Esedoglu, F. Park, and A. Yip. Recent developments in total variation image restoration. In *Mathematical Models in Computer Vision: The Handbook*. 2004.
14. T. F. Chan, G. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Sci. Comput.*, 20:1964–1977, 1999.
15. T. F. Chan, A. Marquina, and P. Mulet. Second order differential functionals in total variation-based image restoration. Technical Report 98-35, UCLA, 1998.
16. T. F. Chan, H. M. Zhou, and R. Chan. Continuation method for total variation denoising problems. Technical Report 95-28, UCLA, 1995.
17. Q. Chang and I.-L. Chern. Acceleration methods for total variation based image denoising. *SIAM J. Sci. Comput.*, 25:982–994, 2003.
18. K. Chen. *Matrix Preconditioning Techniques and Applications*, volume 19 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, UK, 2005.
19. K. Chen and X.-C. Tai. A nonlinear multigrid method for curvature equations related to total variation minimization. Technical Report 05-26, UCLA, 2005.
20. Y. Chen, S. Levine, and M. Rao. Variable exponent, linear growth functionals in image restoration. *SIAM J. Appl. Math.*, 66(4):1383–1406, 2006.
21. Y. Chen, S. Levine, and J. Stanich. Image restoration via nonstandard diffusion. Technical Report 04-01, Duquesne Univ. Dept. of Math and Comp. Sci., 2004.
22. C. Frohn-Schauf, S. Henn, and K. Witsch. Nonlinear multigrid methods for total variation denoising. *Comput. Vis. Sci.*, 7:199–206, 2004.
23. V.-E. Henson. Multigrid methods for nonlinear problems: An overview. Technical report, Center for Applied Scientific Computing Lawrence Livermore Laboratory.
24. K. Ito and K. Kunisch. An active-set strategy based on the augmented lagrangian formulation for image restoration. *M2AN Math. Model. Numer. Anal.*, 33:1–21, 1999.
25. K. Ito and K. Kunisch. BV-type regularization methods for convoluted objects with edge flat and grey scales. *Inverse Probl.*, 16:909–928, 2000.
26. K. Joo and S. Kim. Pde-based image restoration, i: Anti-staircasing and anti-diffusion. Technical report, University of Kentucky, 2003.

27. T. Karkkainen and K. Majava. Nonmonotone and monotone active-set methods for image restoration. *J. Optim. Theory Appl.*, 106(1):61–105, 2000.
28. T. Karkkainen and K. Majava. Semi-adaptive optimization methodology for image denoising. *IEE Proc. Vis. Image Signal Process.*, 152(1):553–560, 2005.
29. T. Karkkainen, K. Majava, and M. Makela. Comparisons of formulations and solution methods for image restoration problems. Technical Report B 14/2000, Department of Mathematical Information Technology University of Jyväskylä, 2000.
30. S-H. Lee and J. K. Seo. Noise removal with gauss curvature driven diffusion. *IEEE Trans. Img. Process.*, 2005.
31. S. Levine, M. Ramsey, T. Misner, and S. Schwab. An adaptive model for image decomposition. Technical Report 05-01, Duquesne Univ. Dept. of Math and Comp. Sci., 2005.
32. M. Lysaker, A. Lundervold, and X.-C. Tai. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *IEEE Trans. Image Process.*, 12, 2003.
33. M. Lysaker and X.-C. Tai. Interactive image restoration combining total variation minimization and a second order functional. *Int. J. Comp Vis.*
34. A. Marquina and S. Osher. Explicit algorithms for a new time dependant model based on level set motion for nonlinear deblurring and noise removal. *SIAM J. Sci. Comput.*, 22:387–405, 2000.
35. S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation based image restoration. *Multiscale Model. and Simul.*, 4:460–489, 2005.
36. S. Osher, A. Sole, and L. Vese. Image decomposition and restoration using total variation minimization and the h^{-1} norm. *Multiscale Model. and Simul.*, 1:349–370, 2003.
37. L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
38. J. Savage and K. Chen. An improved and accelerated nonlinear multigrid method for total-variation denoising. *Int. J. Comput. Math.*, 82:1001–1015, 2005.
39. U. Trottenberg, C. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, London, 2001.
40. L. Vese and S. Osher. Modelling textures and with total variation minimization and oscillating patterns in image processing. Technical Report 02-19, UCLA, 2002.
41. C. Vogel. A multigrid method for total variation based image denoising. In *Computation and Control IV*. Birkhauser, 1995.
42. C. Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, 2002.
43. C. Vogel and M. Oman. Iterative methods for total variation denoising. *SIAM J. Sci. Comput.*, 17:227–238, 1996.
44. T. Washio and C. Oosterlee. Krylov subspace acceleration for nonlinear multigrid schemes. *Electron. Trans. Numer. Anal.*, 6:271–290, 1997.
45. P. Wesseling. *An Introduction to Multigrid Methods*. Wiley, Chichester, 1992.
46. W. Yin, D. Goldfarb, and S. Osher. Image cartoon-texture decomposition and feature selection using the total variation regularized l^1 functional. Technical Report 05-47, UCLA, 2005.

A Method for Total Variation-based Reconstruction of Noisy and Blurred Images

Qianshun Chang¹, Weicheng Wang², and Jing Xu¹

¹ Institute of Applied Mathematics, Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences, Beijing, China. E-mail: {qschang, jingxu}@amss.ac.cn

² Department of Mathematics, National Tsing-Hua University, Hsinchu, Taiwan. E-mail: wangwc@duet.am.nthu.edu.tw

Summary. In this paper, we focus on deblurring and denoising problems for blurred images with moderate or large noise. A new algorithm for the discretized system is presented. Convergence of outer iteration is efficiently improved by adding a linear term on both sides of the system of nonlinear equations. In inner iteration, an algebraic multigrid (AMG) method is applied to solve the linearized systems of equations. We adopt the Krylov subspace method to accelerate the outer nonlinear iteration. Numerical experiments demonstrate that this method is efficient and robust even for images with large noise-to-signal ratios and signal to blurring quantity ratios.

Key words: Image restoration, total variation, nonlinear iteration, algebraic multigrid method, Krylov acceleration

1 Introduction

Image restoration is a fundamental problem in both image processing and computer vision with numerous applications. The blurring of images often occurs from the motion of objects, calibration errors with imaging devices or from unfocused cameras. Main tasks of the image restoration are to recover a “true” image from noisy and blurred data. Mathematically, the image restoration can be written as

$$z = Ku + n, \tag{1}$$

where z is the observed image, u is the true image, K is a known linear blur operator and n is a Gaussian white noise.

In recent years, a popular method for noise removal and deblurring is the total variation based restoration method, proposed by Rudin, Osher and

Fatemi [21]. In this method, the total variation of u is used as a regularization penalty functional for the corresponding minimization problem (see (2) below).

Using the Tikhonov penalty method and a diffusion regularization, the total variation based restoration method can be formulated as an unconstrained minimization problem:

$$\min_u \left(\alpha \int_{\Omega} \sqrt{|\nabla u|^2 + \beta} \, dx dy + \frac{1}{2} \|Ku - z\|_{l^2}^2 \right). \quad (2)$$

Here, $\alpha > 0$ is the penalty parameter and $\beta > 0$ is the regularization parameter and is typically small. The functional in (2) is strictly convex with a unique global minimizer. The well-posedness of problem (2) with $\beta \rightarrow 0+$ has been discussed in [1].

The corresponding Euler-Lagrange equation for (2) is

$$-\alpha \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) + K^*(Ku - z) = 0, \quad (3)$$

where K^* is the adjoint operator of K with respect to standard L_2 inner product.

Various methods have been proposed in literatures to solve (3). For example, a time-marching scheme was suggested in [4, 21]. There the solution of (3) is obtained by evolving the corresponding parabolic equation to steady state. An affine scaling algorithm was proposed in [16]. Vogel and Oman [25] applied a fixed point method to solve equation (3). Newton method with a continuation procedure on the regularization parameter β was used in [9]. Chan, Golub, and Mulet [10] proposed a nonlinear primal-dual method. A multigrid method was proposed to solve the linearization part of equation (3) in [24, 19]. Chang and Chern applied algebraic multigrid method, Krylov subspace algorithm and extrapolation of initial data to accelerate convergence in [14]. However, accuracy and convergence of the algorithms in these papers are only verified for the denoising problems (i.e., the blur operator K is assumed to be the identity operator I).

On the other hand, the recovery of blurred images (generic K) with weak noise (small α) is also quite well understood. In [17], the authors use discrete cosine transform to recover blurred images without noise. In [26], Vogel and Oman present a combination method for the image restoration, they combine the fixed point iteration to handle nonlinearity and the preconditioned conjugate gradient iteration for large linear systems. A noisy and blurred satellite image with very small regularization parameter $\alpha = 5 * 10^{-8}$ is then successfully recovered. Chan, Chan and Wong apply the cosine transform based preconditioner to total variation deblurring and consider strong blur operator and the small regularization parameters $\alpha \in [10^{-2}, 10^{-6}]$ in [8]. In [3], a new modular solver for the image restoration problems is presented and model problems with Gaussian blur and small amount of additive noise are computed.

Our main object in this paper is to develop an efficient and robust solver for (8) over a wide range of the parameter α . In general, the problem (3) is more difficult for moderate or large noise, i.e. larger values of the α , as nonlinear effect is more dominant in this range of α . We present here a new algorithm for solving discretization of the problem (3). Convergence of outer iteration is efficiently improved by adding a linear term γu on both sides of the system of nonlinear equation. In inner iteration, an AMG method is applied to solve the linearized systems of equations. Since the functional (3) is convex, we also adopt the Krylov subspace method [6, 14, 20] to accelerate the outer nonlinear iteration. Numerical experiments demonstrate that our method is efficient and robust for the blurred images with moderate or large noise.

This paper is organized as follows. Section 2 describes the idea and the new method. In Section 3, we briefly explain the AMG algorithm and Krylov subspace acceleration. The model problems and the blur operators described given in Section 4. Finally, numerical results and discussion are given in Section 5.

2 Idea and New Method

Let us consider the Euler-Lagrange equation

$$-\alpha \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) + K^*(Ku - z) = 0 \quad \text{in } \Omega = (0, 1) \times (0, 1), \quad (4)$$

with zero Neumann (no flux) boundary condition. We partition the domain $(0, 1) \times (0, 1)$ into $L \times L$ uniform cells. Denote $1/L$ by h . The cell centers are $(x_l, y_k) = ((l - 1/2)h, (k - 1/2)h)$, $l, k = 1, \dots, L$. The value $u(x_l, y_k)$ is approximated by $u_{l,k}$. Following [25, 14], we discretize (3) by a standard five-point finite difference scheme:

$$\begin{aligned} & \frac{\alpha}{h^2} [(D_{l+1/2,k} + D_{l-1/2,k} + D_{l,k+1/2} + D_{l,k-1/2}) u_{l,k} \\ & - D_{l+1/2,k} u_{l+1,k} - D_{l-1/2,k} u_{l-1,k} - D_{l,k+1/2} u_{l,k+1} - D_{l,k-1/2} u_{l,k-1}] \\ & + (K^*(KU - Z))_{l,k} = 0, \quad l, k = 1, \dots, L, \end{aligned} \quad (5)$$

where

$$D_{l+1/2,k} = \frac{1}{\sqrt{|(u_{l+1,k} - u_{l,k})/h|^2 + \beta}}, \quad (6)$$

and $U = (u_{1,1}, u_{1,2}, \dots, u_{1,L}, u_{2,1}, \dots, u_{2,L}, \dots, u_{L,L})$, $Z = (z_{1,1}, z_{1,2}, \dots, z_{1,L}, z_{2,1}, \dots, z_{L,L})$. The discrete Neumann boundary conditions are

$$u_{0,k} = u_{1,k}, \quad u_{L+1,k} = u_{L,k}, \quad u_{l,0} = u_{l,1}, \quad u_{l,L+1} = u_{l,L}. \quad (7)$$

We abbreviate (5) by

$$\alpha L(U)U + K^*(KU - Z) = 0. \quad (8)$$

In (8), $L(u)$ is strongly nonlinear with wildly varying coefficients. Moreover, the matrix K^*K is full and spectrum of the matrices $L(U)$ and K^*K are differently distributed. See Figure 1 (quoted from [7]) below. As a result,

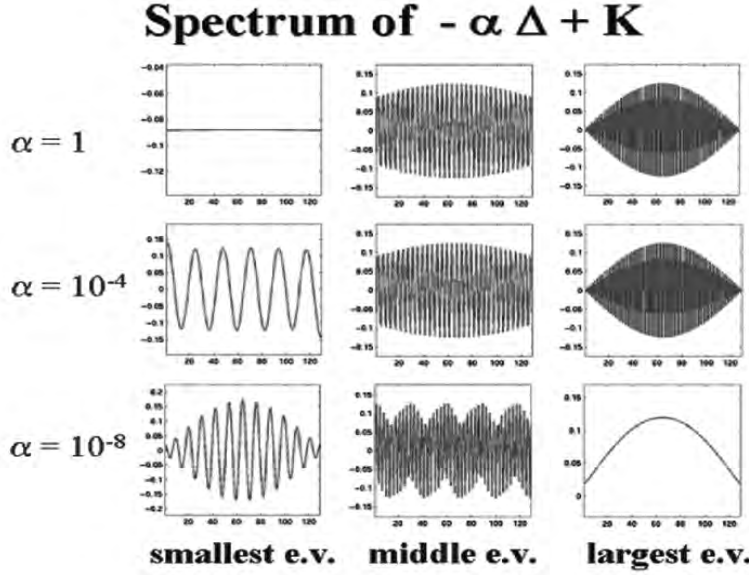


Fig. 1. Spectrum of $-\alpha\Delta + K$

it is not quite easy to solve the nonlinear system (8) by Newton's method efficiently. In [26], the authors combined a fixed point iteration and a product PCG iteration to handle the nonlinear term and the linear system respectively. Another preconditioner based on cosine transform is proposed by Chan, Chan and Wong in [8].

Before stating our algorithm, we first remark on the following approach which seems natural at first sight:

$$\alpha L(u^s)u^{(s+1)} = -K^*(Ku^{(s)} - z). \quad (9)$$

Contrary to intuition, the algorithm (9) may even diverge for large α and weak K , e.g., $\alpha = 10$ and K corresponds to the mask $\frac{1}{64}(1, 1, 4, 1, 1)^T$ (1, 1, 4, 1, 1).

As a first remedy, we add to the matrix $L(u^s)$ a diagonal part. The resulting algorithm is given by

$$(\alpha L(u^{(s)}) + D)u^{(s+1)} = -(K^*K - D)u^{(s)} + K^*z, \quad (10)$$

where D is the diagonal part of the matrix K^*K .

The algorithm (10) turns out to converge only for large α and weak blur operators. This is insufficient for many practical applications. Along this way,

we further increase values of the diagonal entries in (10) by adding the term γu . The new algorithm in this paper is the following:

$$(\alpha L(u^{(s)}) + D + \gamma I)u^{(s+1)} = -(K^*K - D - \gamma I)u^{(s)} + K^*z. \quad (11)$$

The parameter γ is chosen according to the strength of the blur operator K . The larger value of the parameter γ is taken for stronger blur operator K (the strength of K is measured by (22) below).

3 Algorithms for Solving the Nonlinear System of Equations (11)

In the system (11), the unknown u in coefficients of the nonlinear operator L is taken as the previous value $u^{(s)}$. Thus, we apply an outer iteration to solve the nonlinear system (11) and an inner iteration to get $u^{(s+1)}$ for known $u^{(s)}$.

The algebraic multigrid (AMG) method (see[11, 13, 14]) is used as inner iteration.

3.1 Basic AMG Algorithm

Now, we describe our version of the AMG algorithm [11, 13] briefly. We consider the following $n \times n$ system of linear equations

$$AU = F. \quad (12)$$

The AMG method is designed to utilize the principle of the geometrically oriented multigrid (GMG) method to obtain a fast and automatic solution procedure for matrix computations.

In a standard multigrid process, one needs to define the coarse grids, the interpolation operator I_{m+1}^m , the restriction operator I_m^{m+1} , and the coarse grid operator A^{m+1} . The methods differ from each other depending on the choice of the interpolation operators and different algorithms to construct the coarse grid equations and the restriction operators.

We shall adopt Galerkin type algorithm, where $I_m^{m+1} = (I_{m+1}^m)^T$ and $A^{m+1} = I_m^{m+1}A^m I_{m+1}^m$. Thus, we will only need to define the coarse grids and interpolation operators. We follow the approach in [11, 22] to define the grid Ω^m and its coarse grid C^m .

The interpolation operator I_{m+1}^m maps data on Ω^{m+1} to data on Ω^m . Roughly speaking, this interpolation formula is derived so that the i^{th} equation

$$a_{i,i}^m e_i^m + \sum_{j \in N_i^m} a_{i,j}^m e_j^m = r_i^m \approx 0 \quad (13)$$

is almost satisfied. Here, r^m is the residual, $N_i^m = \{j \in \Omega^m \mid a_{i,j}^m \neq 0, j \neq i\}$, which can be thought as the neighbors of i .

In order to solve (13) approximately, we classify the neighbors of the point i into two classes. A point $j \in N_i^m$ is said to be strongly connected to i if

$$|a_{i,j}^m| \geq \theta \cdot \max_{k \neq i} |a_{i,k}^m|$$

for some fixed $0 < \theta \leq 1$, and weakly connected if otherwise. We denote the collection of these neighboring points by S_i^m (strong) and W_i^m (weak), respectively. We also denote $C^m \cap S_i^m$ by C_i^m . Our goal is to derive an interpolation formula

$$e_i^m = \sum_{j \in C_i^m} \omega_{i,j} e_j^m, \text{ for } i \in F^m$$

so that the i^{th} correction equation is almost satisfied:

$$a_{i,i}^m e_i^m + \sum_{j \in N_i^m} a_{i,j}^m e_j^m = 0. \quad (14)$$

Then, we introduce the following geometric assumptions. Two geometrical assumptions are introduced in which extrapolation and averaging formulae are taken into account in the interpolation process.

- (G1) Elements in N_i^m are the neighbors of a point i in Ω^m . Further, the larger the quantity $|a_{i,j}^m|$ is, the closer the point j is to the point i .
- (G2) If $a_{i,j}^m < 0$ or $|a_{i,j}^m|$ is small, we say that the error between i and j is geometrically smooth. Otherwise, we call it geometrically oscillating. Here, we have normalized $a_{i,i} > 0$.

Roughly speaking, “geometrically”, the average location of points in $C^m \cap S_i^m \cap S_j^m$ is somewhere between i and j . Therefore the error e_j^m can be approximated more accurately by an extrapolation formula using e_i and $\sum_{k \in C_i^m \cap S_j^m} g_{j,k} e_k^m$. More precisely, let us define

$$\zeta_{i,j}^m = \frac{-\sum_{k \in C_i^m \cap N_j^m} a_{j,k}^m}{\sum_{k \in C_i^m \cap N_j^m} |a_{j,k}^m|}, \quad \eta_{i,j}^m = \frac{|a_{i,j}^m|}{\frac{1}{|C_i^m \cap N_j^m|} \sum_{k \in C_i^m \cap N_j^m} |a_{j,k}^m|}.$$

The quantity $\zeta_{i,j}^m$ indicates whether there is a large negative entry $a_{j,k}^m$ for $k \in C_i^m \cap N_j^m$. When $\zeta \geq 1/2$ and $a_{i,j}^m < 0$, it can be shown that the errors between the point i and the point j are geometrically smooth. The quantity $\eta_{i,j}^m$ roughly gives the “inverse ratio” of the distance between j and i to the average distance between the point j and the points in $C_i^m \cap N_j^m$. If $\eta_{i,j}^m < 3/4$, we think the “average location” of the points in $C_i^m \cap N_j^m$, denoted by $\bar{k}_{j,i}$, is closer to j than that of i . That is, $\bar{k}_{j,i}$ lies between i and j , and thus, an extrapolation formula for e_j in terms of e_i and $\sum_{k \in C_i^m \cap N_j^m} g_{j,k} e_k^m$ can be applied. When $\eta_{i,j}^m > 2$, we think i is closer to j than that of $\bar{k}_{j,i}$. In this case, we use an interpolation formula instead. Otherwise, we think $\bar{k}_{j,i}$ is

very close to j and we should just use the average formula $\sum_{k \in C_i^m \cap N_j^m} g_{j,k} e_k^m$ to approximate e_j .

In summary, we use the following “geometric” interpolation formulae.

(1) For $j \in S_i^m \cap F^m$, we have

$$e_j^m = \begin{cases} 2 \sum_{k \in C_i^m} g_{j,k} e_k^m - e_i^m, & \text{if } \eta_{i,j}^m < 3/4, \zeta_{i,j} \geq 1/2 \text{ and } a_{i,j}^m < 0, \\ \frac{1}{2} (\sum_{k \in C_i^m} g_{j,k} e_k^m + e_i^m), & \text{if } \eta_{i,j}^m > 2, \zeta_{i,j} \geq 1/2 \text{ and } a_{i,j}^m < 0, \\ \sum_{k \in C_i^m} g_{j,k} e_k^m, & \text{otherwise.} \end{cases} \quad (15)$$

(2) For $j \in W_i^m$, we have

$$e_j^m = \begin{cases} e_i^m, & \text{if } C_i^m \cap S_j^m = \phi, a_{i,j}^m < 0, \\ -e_i^m, & \text{if } C_i^m \cap S_j^m = \phi, a_{i,j}^m > 0, \\ 2 \sum_{k \in C_i^m} g_{j,k} e_k^m - e_i^m, & \text{if } C_i^m \cap S_j^m \neq \phi, \zeta_{i,j} \geq 1/2 \text{ and } a_{i,j}^m < 0, \\ \sum_{k \in C_i^m} g_{j,k} e_k^m, & \text{otherwise.} \end{cases} \quad (16)$$

The convergence proof for this improved AMG method was given in [11, 13] when A^m is symmetric positive definite. Many numerical examples also support the improvement of this “geometric” interpolation formula [11, 13].

3.2 Krylov Subspace Acceleration

The Krylov subspace method [6, 14, 20] is an acceleration technique for general iteration methods. Basically, it uses extrapolation to increase the convergence rate. It is particularly suitable for the outer nonlinear iteration since the functional (2) is convex, or equivalently, the operator of the corresponding Euler-Lagrange equation (3) is monotone. We illustrate this acceleration procedure below. First, we choose two integers M and s , with $M \leq s$. The Krylov subspace acceleration is performed every s steps of outer nonlinear iterations as follows. For integer $n > 0$, let

$$U^{new}(c_1, \dots, c_M) = U^{ns} + \sum_{m=1}^M c_m (U^{ns+1-m} - U^{ns-m}), \quad (17)$$

and minimize the residual of U^{new} with respect to the parameters $c_1, \dots, c_M$ to get

$$\min_{c_1, \dots, c_M} \|Re^{new}(c_1, \dots, c_M)\|_{l^2} = \|Re^{new}(c_1^*, \dots, c_M^*)\|_{l^2}. \quad (18)$$

We then reset U^{ns} to $U^{new}(c_1^*, \dots, c_M^*)$.

Notice that

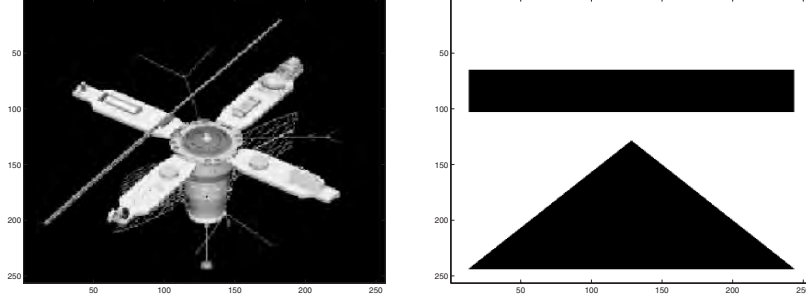


Fig. 2. Original images of Model I (left) and Model II (right)

$$Re^{new} = Re^{ns} + \sum_{m=1}^M c_m (Re^{ns+1-m} - Re^{ns-m}), \quad (19)$$

the coefficients $(c_1, \dots, c_M)$ can be found easily. For instance, if $M = 1$, we have

$$c_1^* = \frac{-\langle Re^{ns}, Re^{ns} - Re^{ns-1} \rangle}{\langle Re^{ns} - Re^{ns-1}, Re^{ns} - Re^{ns-1} \rangle}. \quad (20)$$

4 Models and Blur Operators

In this section, we describe two models and three blur operators used for numerical experiments below. The data source is a satellite image (model I) and a benchmark model problem (model II) used in many papers (for example, [8]). For both models, the original image contains $256 * 256$ pixels. Each pixel is assigned an integer value in $[0, 255]$. The two model images are given in Figure 2.

We consider restoring the two model images blurred by the following three blur operators (see [17]).

- (1) A blurring operator given by the mask

$$\frac{1.2}{784} (1, 2, 3, 16, 3, 2, 1)^T (1, 2, 3, 16, 3, 2, 1).$$

In this model, row sums of the blur matrix are larger than 1.

- (2) An out-of-focus blur,

$$h_{i,j} = \begin{cases} c, & \text{if } |i|, |j| \leq 2, \\ 0, & \text{otherwise,} \end{cases}$$

where $h_{i,j}$ is the j th entry of the first column of the i th block Toeplitz matrix and c is the normalization constant such that $\sum_{i,j} h_{i,j} = 1$.

(3) A truncated Gaussian blur given by

$$h(x, y) = \begin{cases} ce^{-\tau(x^2+y^2)}, & \text{if } |x|, |y| \leq \frac{1}{32}, \\ 0, & \text{otherwise,} \end{cases}$$

here the strength of the blur depends on parameters τ, c (see (22)). Stronger blurs correspond to smaller values of τ or larger values of c . In our computation, $C = 0.01$ and $\tau = 0.1$.

We remark that the blur operator (1) and the Gaussian blur (3) are symmetric and separable, whereas the out-of-focus blur (2) is symmetric but not separable.

5 Numerical Experiments and Discussions

In this numerical experiment, we focus on the performance of different choices of α for the three blur operators mentioned above. Here, 256 is the number of pixels in one direction, i.e., the blur matrix K is of size 256^2 -by- 256^2 . A Gaussian distribution with mean 0 and variance σ is added to the blurred images.

In all computations, we take $\beta = 0.0001$ and $\gamma = 1.0$. We also test $\beta = 1.0e-12$, we find that there is no difference of changing the β for the numerical results and convergence. γ in this paper is just devoted to guarantee the convergence of the algorithm. While for our experience, $\gamma = 1.0$ will meet many situations. We do not need to adjust the γ commonly.

We will use the following signal to noise ratio (SNR) to measure the level of noise

$$SNR = \frac{\|Ku - z\|_{l^2}}{\|u\|_{l^2}}. \quad (21)$$

The signal to blurring quantity ratio

$$SBR = \frac{\|Ku - u\|_{l^2}}{\|u\|_{l^2}}. \quad (22)$$

5.1 Normalized Residual

An important issue in image restorations is to choose a quantity to measure the quality of improvement. It is used as a stopping criterion for the outer nonlinear iteration. Usually, the residual of the system (11) is chosen. However, a normalization is needed in practice. Indeed, we take $D^{-1}(Re)$ as the normalized residual. Here (Re) is the residual of the system (11) and D is the corresponding diagonal matrix. The reason for this normalization is the following. Since the diffusion coefficient is very large in smooth regions, we observe that the unnormalized residual is very large in those components where u is smooth (thus, no more denoising is needed), and is relatively small in

those where u is less smooth (thus, either it has a jump or it needs further denoising). The normalization $D^{-1}(Re)$ therefore takes into account this imbalance. Numerical experiments below demonstrate that this quantity is able to measure the improvement of the denoising and deblurring process. From now on, we shall denote this normalized residual by Re .

5.2 Inner Iteration with AMG Method

In every outer nonlinear iteration, a linearized system of equations needs to be solved. We use the AMG method to solve the system. In the AMG procedure, we apply the simple V-cycle and use the Gauss-Seidel iteration as the smoother. In each outer iteration, only one V-cycle of the AMG method is applied for solving the corresponding linearized system. There is no need to have more inner iterations since the dominant error comes from the outer iteration. The stopping criterion for the outer iteration in this paper is a relative decrease of the (normalized) residual by a factor of 10^{-4} for the blur operator (1), (2) and of 10^{-3} for the blur operator (3). Namely,

$$\frac{\|D^{-1}(Re^N)\|_{l_2}}{\|D^{-1}(Re^0)\|_{l_2}} \leq \begin{cases} 10^{-4}, & \text{for blur operator (1), (2),} \\ 10^{-3}, & \text{for blur operator (3).} \end{cases}$$

The convergence factors of the AMG method in every outer iteration are given in Table 1.

Table 1. ρ_A , the convergence factor of the AMG method in each outer iteration. This result is for the Model I and the blur (2) with $\alpha = 0.1$.

iteration step	1	2	3	4	5	6	7	8
ρ_A	0.071	0.107	0.071	0.064	0.093	0.044	0.044	0.047
iteration step	9	10	11	12	13	14	15	16
ρ_A	0.043	0.045	0.046	0.048	0.055	0.053	0.059	0.062

5.3 Outer Iteration Improved by the Krylov Acceleration

The slow convergence of the outer nonlinear iteration can be improved by the Krylov acceleration method. In the application of Krylov acceleration, we choose the parameter $s = 4$, i.e. we apply the Krylov acceleration every four outer nonlinear iterations. The parameter M is taken to be 1 or 2. The result is given in Table 2. The total number of iterations is reduced to about 50% for slowly convergent blur (1) and about 33% for fast convergent blur (2). In general, Krylov method with $M = 2$ is better than one with $M = 1$. The overhead for Krylov method is low, as only simple algebraic operations are needed. The results demonstrate that the Krylov acceleration method is very

efficient to accelerate the convergence of our outer nonlinear iterations. Unless otherwise specified, we have used Krylov acceleration method with $s = 4$ and $M = 2$ in other examples.

Table 2. Number of outer nonlinear iterations N needed with Krylov acceleration. $M = 0$ means that the Krylov acceleration is not used.

Blur Model		M	N	CPU time (in second)
1	I	0	38	51.01
		1	20	27.45
		2	17	23.39
	II	0	37	48.56
		1	19	25.60
		2	17	22.98
2	I	0	24	32.20
		1	19	26.15
		2	16	22.00
	II	0	23	30.31
		1	16	21.59
		2	16	21.66

5.4 Denoised and Deblurred Results for Images with Large SNR and SBR

Finally, we test the two models for several noising and blurring images. In most computations, our new algorithm is efficient and robust. For the strong blur operator (3), two computational examples are shown in Figure 3 and Figure 4.

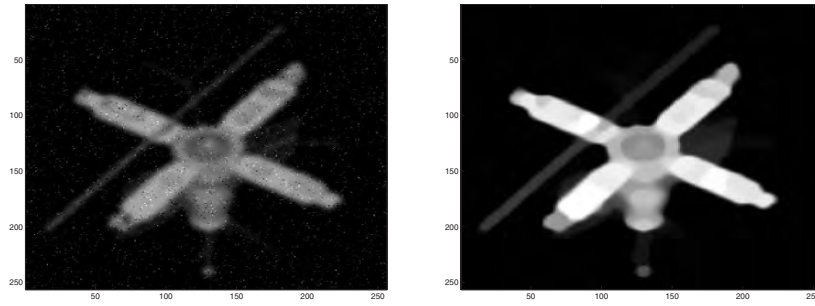


Fig. 3. Noised and blurred (left), and Restored (right) images of Model I and blur operator (3), $\alpha = 0.01$, $SNR = 7.72\%$ and $SBR = 72.16\%$.

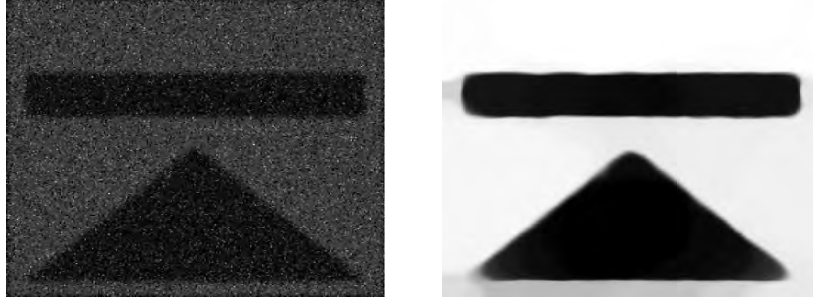


Fig. 4. Noised and blurred (left), and Restored (right) images of Model II and blur operator (3), $\alpha = 0.1$, $SNR = 14.96\%$ and $SBR = 69.63\%$.



Fig. 5. Noised and blurred image (left), Restored (middle) image with $\alpha = 0.1$, and Restored (right) image with $\alpha = 1.0$ for Model II and blur operator (2), $SNR = 22.79\%$ and $SBR = 45.46\%$.

In general, the choice of the parameter α is important in denoising and deblurring problems. Large α is taken when the noise is strong and there is no small construction in the image. We have to take small values of the α if the noise is weak or there is small constructions in the images. Figure 5 shows a larger α is necessary for restoring a image with large SNR.

In conclusion, our numerical experiments demonstrate that the new algorithm is efficient and robust for a wide range of the parameter α .

Remark 1. Here comes a remark, we will consider further choice of the parameter γ and combinational choice of γ with α .

Acknowledgement. The authors thank Professor Tony Chan for his valuable suggestion and help.

References

1. R. Acar and C. R. Vogel, Analysis of total variation penalty methods for ill-posed problems, *Inverse Problems*, 10(1994), pp 1217-1229.

2. L. Alvarez, P. -L. Lions, and J. -M. Morel, Image selective smoothing and edge detection by nonlinear diffusion II, *SIAM J. Numer. Anal.*, 29(1992), pp 845-866.
3. P. Blomgren and T. Chan, Modular solvers for image restorations problems using the discrepancy principle, *Numer. linear Algebra Appl.* Vol. 9 (2002) pp 347-358.
4. C. A. Z. Barcelos, and Y. Chen, Heat flow and related minimization problem in image restoration, *Computers and Mathematics with Applications*, 39(2000), pp 81-97.
5. G. Barles, and P. E. Souganidis, Convergence of approximation schemes for fully nonlinear second order equations, *Asymptotic Analysis* 4(1991), pp 271-283.
6. A. Brandt, and V. Mikulinsky, On recombining iterants in multigrid algorithms and problems with small islands, *SIAM Sci. Comput.*, 16(1995), pp 20-28.
7. T. Chan, Fast solution of nonlinear PDEs arising from image processing, Invited talk of conference "Schnelle loser fur partielle Differnent: algeichungan", oberwolfach, May 28-June 1, 2001.
8. R. Chan, T. Chan and C. Wang, Cosine transform based preconditioners for total variation deblurring, *IEEE Trans. Image processing*, Vol. 8 (1999) pp1472-1478.
9. R. Chan, T. Chan, and H. Zhou, Advanced signal processing algorithms, In proceedings of the International Society of Photo-Optical Instrumentation Engineers, F. Luk, ed., SPIE, 1995, pp 314-325.
10. T. F. Chan, G. H. Golub, and P. Mulet, A nonlinear primal-dual method for total variation-based image restoration, *SIAM J. Sci. Comput.*, 20(1999), pp 1964-1977.
11. Q. Chang, Yaushu Wong and H. Fu, On the algebraic multigrid method, *J. Comput. Phys.*, 125(1996), pp 279-292.
12. Q. Chang, S. Ma and G. Lei, Algebraic multigrid method for queuing networks, *Int. J. of Computer Math.*, 70(1999), pp 539-552.
13. Q. Chang and Z. Huang, Efficient algebraic multigrid algorithms and their convergence, *SIAM J. Sci. Comput.* 24 (2002) pp 597-618.
14. Q. Chang and I. Chern, Acceleration methods for total variation-based image denoising, *SIAM J. Sci. Comput.* 25 (2003) pp 983-994.
15. V.E. Henson and P.S. Vassilevski, Element-free AMGe: General algorithms for computing interpolation weights, *SIAM J. Sci. Comput.* 23 (2001) pp 629-650.
16. Y. Li and F. Santosa, An affine scaling algorithm for minimizing total variation in image enhancement, Tech. Report 12/94, Center for Theory and Simulation in Science and Engineering, Cornell University, 1994.
17. M. Na and R. Chan and W. Tang, A fast algorithm for deblurring models with Neumann boundary conditions, *SIAM. J. Sci. Comput.*, Vol. 22, (1999) pp 851-866.
18. S. McCormick, *Multigrid methods*, SIAM, Philadelphia, 1987.
19. M. E. Oman, Fast multigrid techniques in total variation-based image reconstruction, In proceedings of the 1995 Copper Mountain Conference on Multigrid Methods, 1995.
20. C. W. Oosterlee, and T. Washio, Krylov subspace acceleration of nonlinear multigrid with application to recirculating flows, *SIAM J. Sci. Comput.*, 21(2000), pp 1670-1690.
21. L. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Phys. D*, 60(1992), pp 259-268.

22. J. Ruge and K. Stüben, Algebraic multigrid, In Multigrid Methods, (S. F. McCormick, ed.) 4, SIAM, Philadelphia, (1987) pp 73-130.
23. P.S. Vassilevski and J. G.Wade, A comparison of multilevel methods for total variation regularization, Electric Transaction on Numerical Analysis, 6 (1997), pp 255-280.
24. C. R. Vogel, A multigrid method for total variation-based image denoising, In Computation and Control IV, Progr. Systems Control Theory 20, Birkhauser, Boston, MA, 1995, pp 323-331.
25. C. R. Vogel, and M. E. Oman, Iterative methods for total variation denoising, SIAM. J. Sci. Comput., 17(1996), pp 227-238.
26. C. R. Vogel and M. E. Oman, Fast, robust total variation-based reconstruction of noisy, blurred images, IEEE Trans. Image processing, Vol.7 (1998) pp 813-824.
27. Y. Zhou, Applications of discrete functional analysis to the finite difference method, International Academic Publishers, Beijing, 1991.

Minimization of an Edge-Preserving Regularization Functional by Conjugate Gradient Type Methods

Jian-Feng Cai¹, Raymond H. Chan^{1*}, and Benedetta Morini^{2**}

¹ Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong. E-mail: {jfc, rchan}@math.cuhk.edu.hk

² Dipartimento di Energetica “S. Steccco”, Università di Firenze, via C. Lombroso 6/17, Firenze, Italia. E-mail: benedetta.morini@unifi.it

Summary. Recently, a powerful two-phase method for removing impulse noise has been developed. It gives a satisfactory result even for images with 90% pixels corrupted by impulse noise. However, the two-phase method is not computationally efficient, because it requires the minimization of a non-smooth functional in the second phase, which is done by a relaxation-based method. In this paper, we remove the non-smooth term from the functional, and call for the minimization of a smooth one. The minimizer is then found by using a conjugate gradient method proposed by J. Sun and J. Zhang. We prove the global convergence of the conjugate gradient type method applied to our functional. Simulation results show that our method is several times faster than the relaxation-based method when the noise ratio is high.

1 Introduction

Impulse noise is caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or transmission in a noisy channel [2]. Let $\mathbf{x}$ denote the original image and $[s_{\min}, s_{\max}]$ denote the dynamic range of $\mathbf{x}$. The impulse noise model with noise ratio (error probability) p for a noisy image $\mathbf{y}$ is

$$y_{i,j} = \begin{cases} r_{i,j}, & \text{with probability } p, \\ x_{i,j}, & \text{with probability } 1 - p, \end{cases}$$

where $x_{i,j}$ and $y_{i,j}$ are the gray levels of the original image $\mathbf{x}$ and the noisy image $\mathbf{y}$ at pixel location (i, j) . There are two main models to represent impulse

*This work was supported by HKRGC Grant CUHK 400503 and CUHK DAG 2060257.

**The research was partially supported by GNCS-INDAM and MIUR Italia through “Cofinanziamenti Programmi di Ricerca Scientifica di Interesse Nazionale”.

noise: the salt-and-pepper noise and the random-valued noise. For images corrupted by salt-and-pepper noise, $r_{i,j}$ can only take values $s_{\min}$ or $s_{\max}$ while for random-valued noise, $r_{i,j}$ can be any identically distributed, independent random number in $[s_{\min}, s_{\max}]$.

There are two popular types of methods for removing impulse noise. One is the median filter and its variants [7, 13]. It can detect the noise pixels accurately but it restores them poorly when the noise ratio is high. The gray levels of uncorrupted pixels are unchanged. The recovered image may lose its details and be distorted. Another procedure, the variational approach, is capable of retaining the details and the edges well but the gray level of every pixel is changed including uncorrupted ones [14].

Recently, a two-phase scheme for removing impulse noise has been proposed in [4, 5]. This scheme combines the advantages of both the median-type filters and the variational approach. In the first phase, a median-type filter is used to identify pixels which are likely to be contaminated by noise (noise candidates). In the second phase, the image is restored by minimizing a specialized regularization functional that applies only to those selected noise candidates. Therefore, the details and edges of the image can be preserved, and the uncorrupted pixels are unchanged.

The two-phase scheme is powerful even for noise ratio as high as 90%, see [4]. However, the functional to be minimized in the second phase is non-smooth, and it is costly to get the minimizer. Here we modify the functional by removing the non-smooth data-fitting term to get a smooth one. Therefore, many sophisticated methods developed for smooth optimization are applicable.

In this paper, conjugate gradient (CG) type methods are applied to minimize the smooth functional. Based on the results in [18], we apply CG methods in which the line search step is replaced by a step whose length is determined by a special formula. We prove that such CG methods are globally convergent for our minimization functional. Simulation results show that when the noise ratio is high, our method is several times faster than the relaxation method used in [4, 5].

The outline of the paper is as follows. In Section 2, we review the method presented in [4, 5]. In Section 3, we present our method. In Section 4, we give the convergence results of the method. In Section 5, simulation results are presented and finally in Section 6 we conclude the paper.

2 Review of Two Phase Methods

In this section we give a brief review on the two-phase method for removing salt-and-pepper impulse noise [4] and random-valued impulse noise [5]. The first phase is the detection of the noise pixels and the second phase is the recovering of the noise pixels detected in the first phase.

The First Phase: Detection of Noise Pixels

The first phase is the detection of the noise pixels. For salt-and-pepper noise, this is accomplished by using the adaptive median filter (AMF) [13] while for random-valued noise, it is accomplished by using the adaptive center-weighted median filter (ACWMF) [7]. Since we are concerned with accelerating the minimization procedure in the second phase, we only consider salt-and-pepper noise in the paper. The method can be applied equally well to random-valued noise.

The Second Phase: Recovering of Noise Pixels

We first give some notations. Let X be an image of size M -by- N and $\mathcal{A} = \{1, 2, 3, \dots, M\} \times \{1, 2, 3, \dots, N\}$ be the index set of the image X . Let $\mathcal{N} \subset \mathcal{A}$ be the set of indices of the noise pixels detected from the first phase and c be its number of elements. Let $\mathcal{V}_{i,j}$ be the set of the four closest neighbors of the pixel at position $(i, j) \in \mathcal{A}$. Let $y_{i,j}$ be the observed pixel value of the image at position (i, j) . In [4], the recovering of noise pixels calls for the minimization of the functional:

$$F_\alpha(\mathbf{u}) = \sum_{(i,j) \in \mathcal{N}} \left[|u_{i,j} - y_{i,j}| + \frac{\beta}{2} (2 \cdot S_{i,j}^1 + S_{i,j}^2) \right], \quad (1)$$

where

$$S_{i,j}^1 = \sum_{(m,n) \in \mathcal{V}_{i,j} \setminus \mathcal{N}} \varphi_\alpha(u_{i,j} - y_{m,n}), \quad (2)$$

$$S_{i,j}^2 = \sum_{(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}} \varphi_\alpha(u_{i,j} - u_{m,n}), \quad (3)$$

φ_α is an edge-preserving function and $\mathbf{u} = [u_{i,j}]_{(i,j) \in \mathcal{N}}$ is a column vector of length c ordered lexicographically. We assume that the edge-preserving function φ_α is: (a) twice continuously differentiable, (b) $\varphi_\alpha'' > 0$, and (c) even. Examples of such $\varphi_\alpha(t)$ are $\sqrt{t^2 + \alpha}$ and $\log(\cosh(\alpha t))$ where $\alpha > 0$ is a parameter, see [6] and [11]. From the above properties, we can conclude that $\varphi_\alpha(t)$ is strictly increasing with $|t|$ and coercive, i.e. $\varphi_\alpha(t) \rightarrow \infty$ as $|t| \rightarrow \infty$.

In [4], (1) is minimized by using a 1-D relaxation method. More precisely, at each iteration, we minimize (1) with respect to only one unknown while all the other unknowns are fixed. The procedure is repeated until convergence. In each iteration, a 1-D nonlinear equation is to be solved. Newton's method with special initial guess that guarantees quadratic convergence is used to solve these nonlinear equations, see [3] for detail.

3 Our Method

The function F_α in (1) is a non-smooth functional because of the $|u_{i,j} - y_{i,j}|$ term — the *data-fitting* term. In our method, we first remove this term. It is motivated by the following two facts:

1. The data-fitting term keeps the minimizer $\mathbf{u}$ close to the original image $\mathbf{y}$ so that the pixels which are uncorrupted in the original image are not altered. However, in the two-phase method the functional F_α is cleaning only the noise pixels and the uncorrupted pixels are unchanged. Hence, the data-fitting term is not required. This fact is verified numerically in [4].
2. Removing the data-fitting term will make F_α to be a smooth functional which can be minimized efficiently.

Therefore, the functional that we are minimizing in this paper is

$$\mathcal{F}_\alpha(\mathbf{u}) = \sum_{(i,j) \in \mathcal{N}} (2 \cdot S_{i,j}^1 + S_{i,j}^2), \quad (4)$$

where $S_{i,j}^1$ and $S_{i,j}^2$ are the same as those defined in (2) and (3). Simulation results in Section 5 show that the minimizers of (1) and (4) attain the same signal-to-noise ratio.

The minimization methods we use to solve (4) is the conjugate gradient (CG) type method proposed in [18]. It does not need the Hessian matrix nor perform the line search. The resulting CG method can find the minimizer more efficiently by avoiding these time consuming tasks. We remark that the Hessian of (4) has not any special structure, so it is difficult to do preconditioning. Therefore, we only consider non-preconditioned CG here. We will give a very brief description of the method here.

The Minimization Algorithm

The general conjugate gradient method applied to $\min_{\mathbf{u}} \mathcal{F}_\alpha(\mathbf{u})$ has the following form. Given $\mathbf{u}_0$, let

$$\mathbf{d}_k = \begin{cases} -\mathbf{g}_k & \text{for } k = 0, \\ -\mathbf{g}_k + \beta_k \mathbf{d}_{k-1} & \text{for } k > 0, \end{cases} \quad (5)$$

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k \mathbf{d}_k, \quad (6)$$

where $\mathbf{g}_k = \nabla \mathcal{F}_\alpha(\mathbf{u}_k)$, α_k is determined by line-search and β_k is chosen so that $\mathbf{d}_k$ is the k -th conjugate direction when the function is quadratic and the line search is exact. Some of the well-known formula for β_k are:

$$\beta_k^{FR} = \frac{\|\mathbf{g}_k\|^2}{\|\mathbf{g}_{k-1}\|^2} \quad (\text{Fletcher-Reeves [10]}), \quad (7)$$

$$\beta_k^{PR} = \frac{\mathbf{g}_k^T(\mathbf{g}_k - \mathbf{g}_{k-1})}{\|\mathbf{g}_{k-1}\|^2} \quad (\text{Polak-Ribière [15]}), \quad (8)$$

$$\beta_k^{HS} = \frac{\mathbf{g}_k^T(\mathbf{g}_k - \mathbf{g}_{k-1})}{\mathbf{d}_{k-1}^T(\mathbf{g}_k - \mathbf{g}_{k-1})} \quad (\text{Hestenes-Stiefel [12]}), \quad (9)$$

$$\beta_k^{CD} = \frac{\|\mathbf{g}_k\|^2}{-\mathbf{d}_{k-1}^T \mathbf{g}_{k-1}} \quad (\text{The Conjugate Descent Method [9]}), \quad (10)$$

$$\beta_k^{DY} = \frac{\|\mathbf{g}_k\|^2}{\mathbf{d}_{k-1}^T(\mathbf{g}_k - \mathbf{g}_{k-1})} \quad (\text{Dai-Yuan [8]}). \quad (11)$$

In [18], it is proved that if $\mathcal{F}_\alpha$ satisfies the following Assumption 1 and α_k is chosen according to a special formula (see (14) below), then the resulting CG method is globally convergent.

Assumption 1

1. Let $\Delta = \{\mathbf{u} \mid \mathcal{F}_\alpha(\mathbf{u}) \leq \mathcal{F}_\alpha(\mathbf{u}_0)\}$. Then there exists a neighborhood Ω of Δ such that $\nabla \mathcal{F}_\alpha$ is Lipschitz continuous on Ω , i.e. there exists a Lipschitz constant $\mu > 0$ such that

$$\|\nabla \mathcal{F}_\alpha(\mathbf{u}) - \nabla \mathcal{F}_\alpha(\mathbf{v})\| \leq \mu \|\mathbf{u} - \mathbf{v}\|, \quad \forall \mathbf{u}, \mathbf{v} \in \Omega, \quad (12)$$

2. $\mathcal{F}_\alpha$ is strongly convex in Ω , i.e. there exists a $\lambda > 0$ such that

$$(\nabla \mathcal{F}_\alpha(\mathbf{u}) - \nabla \mathcal{F}_\alpha(\mathbf{v}))^T (\mathbf{u} - \mathbf{v}) \geq \lambda \|\mathbf{u} - \mathbf{v}\|^2, \quad \forall \mathbf{u}, \mathbf{v} \in \Omega. \quad (13)$$

In that case, we choose $\{Q_k\}$ to be a sequence of c -by- c positive definite matrices such that

$$\nu_{\min} \mathbf{d}^T \mathbf{d} \leq \mathbf{d}^T Q_k \mathbf{d} \leq \nu_{\max} \mathbf{d}^T \mathbf{d}, \quad \forall \mathbf{d} \in \mathbb{R}^c,$$

with $\nu_{\min} > 0$ and $\nu_{\max} > 0$. Then the step length α_k is defined as

$$\alpha_k = -\frac{\delta \mathbf{g}_k^T \mathbf{d}_k}{\mathbf{d}_k^T Q_k \mathbf{d}_k}, \quad \text{where } \delta \in (0, \frac{\nu_{\min}}{\mu}). \quad (14)$$

If $\mathcal{F}_\alpha$ satisfies Assumption 1, the sequence $\{\mathbf{u}_k\}$ defined by (5), (6) and (14) is globally convergent for all choices of β_k in (7) – (11), see [18].

4 Convergence of the Method

The minimization of (4) is a constrained minimization problem as the minimizer must lie in the dynamic range $[s_{\min}, s_{\max}]^c = \{\mathbf{u} \in \mathbb{R}^c : s_{\min} \leq u_i \leq$

$s_{\max}, i = 1, \dots, c\}$. We are going to show that it is indeed a convex unconstrained minimization problem. In fact, we show that the functional $\mathcal{F}_\alpha$ is strictly convex in $\mathbb{R}^c$ and its minimizer lies in $[s_{\min}, s_{\max}]^c$. Moreover, we show that $\mathcal{F}_\alpha$ satisfies Assumption 1, hence the CG method is globally convergent.

To show that $\mathcal{F}_\alpha$ is strictly convex we first derive some properties of the Hessian matrix. As stated before,

$$\mathcal{F}_\alpha(\mathbf{u}) = \sum_{(i,j) \in \mathcal{N}} (2 \cdot S_{i,j}^1 + S_{i,j}^2).$$

Because φ_α is an even function, we get

$$\begin{aligned} (\nabla \mathcal{F}_\alpha(\mathbf{u}))_{(i,j) \in \mathcal{N}} = \\ 2 \sum_{(m,n) \in \mathcal{V}_{i,j} \setminus \mathcal{N}} \varphi'_\alpha(u_{i,j} - y_{m,n}) + 2 \sum_{(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}} \varphi'_\alpha(u_{i,j} - u_{m,n}). \end{aligned}$$

Hence

$$(\nabla^2 \mathcal{F}_\alpha(\mathbf{u}))_{((i,j),(p,q))} = \begin{cases} 2(R_{i,j}^1 + R_{i,j}^2), & \text{if } (i,j) = (p,q), \\ -2\varphi''_\alpha(u_{i,j} - u_{p,q}), & \text{if } (p,q) \in \mathcal{V}_{i,j} \cap \mathcal{N}, \\ 0, & \text{otherwise,} \end{cases} \quad (15)$$

where

$$\begin{aligned} R_{i,j}^1 &= \sum_{(m,n) \in \mathcal{V}_{i,j} \setminus \mathcal{N}} \varphi''_\alpha(u_{i,j} - y_{m,n}), \\ R_{i,j}^2 &= \sum_{(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}} \varphi''_\alpha(u_{i,j} - u_{m,n}). \end{aligned}$$

Consider another matrix $\mathcal{G}_\alpha$ of size MN -by- MN defined by

$$(\mathcal{G}_\alpha)_{((i,j),(p,q))} \triangleq \begin{cases} 2(R_{i,j}^1 + R_{i,j}^2), & \text{if } (i,j) = (p,q) \in \mathcal{N}, \\ 2(T_{i,j}^1 + T_{i,j}^2), & \text{if } (i,j) = (p,q) \notin \mathcal{N}, \\ -2\varphi''_\alpha(y_{i,j} - u_{p,q}), & \text{if } (i,j) \notin \mathcal{N}, (p,q) \in \mathcal{N} \text{ and } (p,q) \in \mathcal{V}_{i,j}, \\ -2\varphi''_\alpha(u_{i,j} - y_{p,q}), & \text{if } (i,j) \in \mathcal{N}, (p,q) \notin \mathcal{N} \text{ and } (p,q) \in \mathcal{V}_{i,j}, \\ -2\varphi''_\alpha(u_{i,j} - u_{p,q}), & \text{if } (i,j) \in \mathcal{N}, (p,q) \in \mathcal{N} \text{ and } (p,q) \in \mathcal{V}_{i,j}, \\ -2\varphi''_\alpha(y_{i,j} - y_{p,q}), & \text{if } (i,j) \notin \mathcal{N}, (p,q) \notin \mathcal{N} \text{ and } (p,q) \in \mathcal{V}_{i,j}, \\ 0, & \text{otherwise,} \end{cases}$$

where

$$\begin{aligned} T_{i,j}^1 &= \sum_{(m,n) \in \mathcal{V}_{i,j} \setminus \mathcal{N}} \varphi''_\alpha(y_{i,j} - y_{m,n}), \\ T_{i,j}^2 &= \sum_{(m,n) \in \mathcal{V}_{i,j} \cap \mathcal{N}} \varphi''_\alpha(y_{i,j} - u_{m,n}). \end{aligned}$$

Then since $\varphi''_\alpha > 0$, $\mathcal{G}_\alpha$ has exactly the same graph as the 2D Laplacian, and thus is irreducible. In addition, $\mathcal{G}_\alpha$ has row sum being zero, except on rows corresponding to pixels on the boundary and in that case the row sum is strictly greater than zero. Hence $\mathcal{G}_\alpha$ is irreducibly diagonally dominant and so by Corollary 1.22 of [19], $\mathcal{G}_\alpha$ is positive definite.

Now, note that $\nabla^2 \mathcal{F}_\alpha(\mathbf{u})$ is a principal sub-matrix of $\mathcal{G}_\alpha$, formed by deleting the rows and columns in $\mathcal{G}_\alpha$ corresponding to the pixels not in $\mathcal{N}$. Thus $\nabla^2 \mathcal{F}_\alpha(\mathbf{u})$ is also positive definite.

We summarize the results below:

Theorem 1. *For any given $\mathbf{u} \in \mathbb{R}^c$, the matrix $\nabla^2 \mathcal{F}_\alpha(\mathbf{u})$ defined in (15) is positive definite, i.e.,*

$$\lambda_{\min}(\nabla^2(\mathcal{F}_\alpha(\mathbf{u}))) > 0,$$

where $\lambda_{\min}(\nabla^2(\mathcal{F}_\alpha(\mathbf{u})))$ is the minimal eigenvalue of $\nabla^2(\mathcal{F}_\alpha(\mathbf{u}))$.

Theorem 2. *The functional $\mathcal{F}_\alpha$ given in (4) has only one local minimum which is also the global minimum. The global minimizer $\mathbf{u}^*$ of $\mathcal{F}_\alpha$ is always within the dynamic range, i.e. $\mathbf{u}^* \in [s_{\min}, s_{\max}]^c$.*

Proof. By Theorem 1, $\mathcal{F}_\alpha$ is strictly convex. Then, a local minimum of $\mathcal{F}_\alpha$ is also a global minimum and there exists at most one global minimum, see Proposition B.10 in [1].

To show that the global minimum exists, consider the box

$$S = \{\mathbf{u} \in \mathbb{R}^c \mid a \leq u_i \leq b, i = 1, \dots, c\}$$

with $a < s_{\min}$ and $s_{\max} < b$. Since S is compact and $\mathcal{F}_\alpha$ is continuous and strictly convex, there exists the global minimizer $\mathbf{u}^* = (u_{i,j}^*)_{(i,j) \in \mathcal{N}}$ of $\mathcal{F}_\alpha$ over S . Now we show that $\mathbf{u}^*$ lies in the interior of S , and hence $\mathbf{u}^*$ is the global minimizer of $\mathcal{F}_\alpha$ over $\mathbb{R}^c$. To this end, note that if $\mathbf{u}^*$ belongs to the boundary of S , then there exists a point $\mathbf{u}$ in the interior of S with $\mathcal{F}_\alpha(\mathbf{u}) < \mathcal{F}_\alpha(\mathbf{u}^*)$. Indeed, we define

$$u_{i,j} = \begin{cases} s_{\max}, & s_{\max} < u_{i,j}^* \leq b, \\ s_{\min}, & a \leq u_{i,j}^* < s_{\min}, \\ u_{i,j}^*, & \text{otherwise.} \end{cases} \quad (16)$$

Then we have

$$\begin{cases} |u_{i,j} - u_{p,q}| \leq |u_{i,j}^* - u_{p,q}^*|, & (p,q) \in \mathcal{V}_{i,j} \cap \mathcal{N}, \\ |u_{i,j} - y_{p,q}| \leq |u_{i,j}^* - y_{p,q}|, & (p,q) \in \mathcal{V}_{i,j} \setminus \mathcal{N}. \end{cases} \quad (17)$$

Since at least one of the $u_{i,j}^*$ is on the boundary of S and all the $y_{p,q}$ are in $[s_{\min}, s_{\max}]$, we can conclude that at least one of the equalities in (17) is a strict inequality. Since $\mathcal{F}_\alpha$ is a sum of terms of the form $\varphi_\alpha(v - w)$ and $\varphi_\alpha(v - w)$ is strictly increasing w.r.t the difference $|v - w|$, $\mathcal{F}_\alpha(\mathbf{u}) < \mathcal{F}_\alpha(\mathbf{u}^*)$. Hence $\mathbf{u}^*$ cannot be the minimizer of (4) over S . Thus the minimizer $\mathbf{u}^*$ must

be in the interior of S , and it is therefore also the global minimizer of $\mathcal{F}_\alpha$ in $\mathbb{R}^c$.

Finally, to show that $\mathbf{u}^* \in [s_{\min}, s_{\max}]^c$, we proceed as above. In particular, if some components of $\mathbf{u}^*$ are outside $[s_{\min}, s_{\max}]$, we define a new point $\mathbf{u}$ as in (16). Then again we will have a contradiction that $\mathcal{F}_\alpha(\mathbf{u}) < \mathcal{F}_\alpha(\mathbf{u}^*)$. $\square$

Theorem 2 shows that the minimization problem can be viewed as an unconstrained minimization problem. Next we show that $\mathcal{F}_\alpha$ satisfies the Assumption 1.

Theorem 3. *Let $\{\mathbf{u}_k\}$ be the sequence generated by the conjugate gradient method. Then, the function $\mathcal{F}_\alpha$ defined in (4) satisfies (12) and (13).*

Proof. Since φ_α is continuous and coercive, $\mathcal{F}_\alpha(\mathbf{u}) \rightarrow \infty$ as $\|\mathbf{u}\| \rightarrow \infty$. To show this, we proceed by contradiction and suppose that $\mathcal{F}_\alpha(\mathbf{u})$ is bounded for $\|\mathbf{u}\| \rightarrow \infty$. Note that if there is one noisy pixel $|u_{i,j}| \rightarrow \infty$ having at least one non-noisy neighbor, then $S_{i,j}^1 \rightarrow \infty$ and consequently $\mathcal{F}_\alpha(\mathbf{u}) \rightarrow \infty$. Therefore, if $\mathcal{F}_\alpha(\mathbf{u})$ is bounded for $\|\mathbf{u}\| \rightarrow \infty$ we conclude that for each noisy pixel $|u_{i,j}| \rightarrow \infty$ all its neighbors are noisy and tend to infinity at the same rate as $|u_{i,j}|$. Repeating this argument for each of such neighbors, we conclude that all the pixels are noisy, i.e. $\mathcal{A} \equiv \mathcal{N}$ which is impossible.

Since $\mathcal{F}_\alpha(\mathbf{u}) \rightarrow \infty$ as $\|\mathbf{u}\| \rightarrow \infty$, given the initial guess $\mathbf{u}_0$, the level set $\Delta = \{\mathbf{u} \mid \mathcal{F}_\alpha(\mathbf{u}) \leq \mathcal{F}_\alpha(\mathbf{u}_0)\}$ must be bounded. Let $(u_0)_{k,l}$ be an arbitrary component of $\mathbf{u}_0$, and

$$z = \max \left\{ |(u_0)_{k,l}|, \max_{(i,j) \in \mathcal{V}_{k,l}} |(u_0)_{(i,j)}| \right\}.$$

Then we define a new vector $\mathbf{w}$ by replacing the entry $(u_0)_{k,l}$ by $w_{k,l} = 1 + 3z$. Then, for any neighbors v of $(u_0)_{k,l}$ we have

$$\begin{aligned} |(u_0)_{k,l} - v| &< 1 + (|v| - v) + |(u_0)_{k,l}| + |v| \\ &= 1 + |(u_0)_{k,l}| + 2|v| - v \leq 1 + 3z - v = |w_{k,l} - v|, \end{aligned}$$

and consequently, $\mathcal{F}_\alpha(\mathbf{u}_0) < \mathcal{F}_\alpha(\mathbf{w})$. Therefore,

$$\Delta \subseteq \Omega \equiv \{\mathbf{u} \mid \mathcal{F}_\alpha(\mathbf{u}) < \mathcal{F}_\alpha(\mathbf{w})\}.$$

By the continuity of $\mathcal{F}_\alpha$, Ω is an open set and its closure is

$$\bar{\Omega} = \{\mathbf{u} \mid \mathcal{F}_\alpha(\mathbf{u}) \leq \mathcal{F}_\alpha(\mathbf{w})\}.$$

Repeating the argument in the first paragraph of this proof, we see that the closure $\bar{\Omega}$ is also bounded. Moreover,

$$\|\nabla^2 \mathcal{F}_\alpha(\mathbf{u})\| \leq \sup_{\mathbf{v} \in \bar{\Omega}} \|\nabla^2 \mathcal{F}_\alpha(\mathbf{v})\| = \max_{\mathbf{v} \in \bar{\Omega}} \|\nabla^2 \mathcal{F}_\alpha(\mathbf{v})\|, \quad \text{for all } \mathbf{u} \in \Omega,$$

since $\nabla^2 \mathcal{F}_\alpha(\mathbf{v})$ is a continuous function of $\mathbf{v}$ on the bounded and closed set $\bar{\Omega}$, hence the supremum can be attained in $\bar{\Omega}$. So by Theorem 9.19 of [16], we have the desired result (12) by taking $\mu = \max_{\mathbf{v} \in \bar{\Omega}} \|\nabla^2 \mathcal{F}_\alpha(\mathbf{v})\|$.

By Taylor's expansion on $\mathcal{F}_\alpha$, we have

$$\mathcal{F}_\alpha(\mathbf{u}) = \mathcal{F}_\alpha(\mathbf{v}) + \nabla \mathcal{F}_\alpha(\mathbf{v})^T (\mathbf{u} - \mathbf{v}) + \frac{1}{2} (\mathbf{u} - \mathbf{v})^T \nabla^2 \mathcal{F}_\alpha(\bar{\mathbf{u}}) (\mathbf{u} - \mathbf{v}), \quad (18)$$

and

$$\mathcal{F}_\alpha(\mathbf{v}) = \mathcal{F}_\alpha(\mathbf{u}) + \nabla \mathcal{F}_\alpha(\mathbf{u})^T (\mathbf{v} - \mathbf{u}) + \frac{1}{2} (\mathbf{v} - \mathbf{u})^T \nabla^2 \mathcal{F}_\alpha(\bar{\mathbf{v}}) (\mathbf{v} - \mathbf{u}), \quad (19)$$

where $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ lie on the line segment with end-points $\mathbf{u}, \mathbf{v} \in \Omega$. Adding up (18) and (19) and rearranging, we have

$$(\nabla \mathcal{F}_\alpha(\mathbf{u}) - \nabla \mathcal{F}_\alpha(\mathbf{v}))^T (\mathbf{u} - \mathbf{v}) = \frac{1}{2} (\mathbf{u} - \mathbf{v})^T (\nabla^2 \mathcal{F}_\alpha(\bar{\mathbf{u}}) + \nabla^2 \mathcal{F}_\alpha(\bar{\mathbf{v}})) (\mathbf{u} - \mathbf{v}).$$

Note that for a positive definite matrix A ,

$$\mathbf{x}^T A \mathbf{x} \geq \lambda_{\min}(A) \mathbf{x}^T \mathbf{x} = \lambda_{\min}(A) \|\mathbf{x}\|^2,$$

where $\lambda_{\min}(A)$ is the smallest eigenvalue of A . Hence, together with the result of Theorem 1, we have:

$$\begin{aligned} & (\nabla \mathcal{F}_\alpha(\mathbf{u}) - \nabla \mathcal{F}_\alpha(\mathbf{v}))^T (\mathbf{u} - \mathbf{v}) \\ & \geq \frac{1}{2} (\lambda_{\min}(\nabla^2 \mathcal{F}_\alpha(\bar{\mathbf{u}})) + \lambda_{\min}(\nabla^2 \mathcal{F}_\alpha(\bar{\mathbf{v}}))) \|\mathbf{u} - \mathbf{v}\|^2 \\ & \geq \frac{1}{2} \cdot 2 \cdot \inf_{\mathbf{z} \in \bar{\Omega}} \lambda_{\min}(\nabla^2 \mathcal{F}_\alpha(\mathbf{z})) \|\mathbf{u} - \mathbf{v}\|^2 \\ & = \lambda \|\mathbf{u} - \mathbf{v}\|^2, \end{aligned}$$

where $\lambda \equiv \inf_{\mathbf{z} \in \bar{\Omega}} \lambda_{\min}(\nabla^2 \mathcal{F}_\alpha(\mathbf{z}))$. Since $\lambda_{\min}(\nabla^2 \mathcal{F}_\alpha(\mathbf{z}))$ is a continuous function of $\mathbf{z}$ on the closed and bounded set $\bar{\Omega}$ (see Corollary 4.10 in [17]), we have $\lambda = \lambda_{\min}(\nabla^2 \mathcal{F}_\alpha(\mathbf{z}_0))$ for some $\mathbf{z}_0 \in \bar{\Omega}$. By Theorem 1, $\lambda > 0$. This proves (13). $\square$

We conclude by providing a global convergence result of the CG method applying to (4).

Theorem 4. *Let $\{\mathbf{u}_k\}$ be the sequence generated by the conjugate gradient method with α_k given in (14). Then, for any choice of β_k in (7)–(11), $\{\mathbf{u}_k\}$ converges to the global minimum of $\mathcal{F}_\alpha$.*

Proof. By Theorem 9 of [18], $\lim_{k \rightarrow \infty} \|\nabla \mathcal{F}_\alpha(\mathbf{u}_k)\| = 0$. Hence, all the limit points of $\{\mathbf{u}_k\}$ are stationary points of $\mathcal{F}_\alpha$. By Theorem 2, the thesis follows. $\square$

5 Simulation

Throughout the simulations, we use MATLAB 7.01 (R14) on a PC equipped with Intel Pentium 4 CPU 3.00 GHz and 1,024 MB RAM memory. Our test images are the 512-by-512 *goldhill* and *lena* images. To assess the restoration performance qualitatively, we use the PSNR (peak signal to noise ratio, see [2]) defined as

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i,j} (x_{i,j}^r - x_{i,j})^2},$$

where $x_{i,j}^r$ and $x_{i,j}$ denote the pixel values of the restored image and the original image respectively.

We emphasize that in this paper, we are concerned with the speed of solving the minimization problem in the second phase of the two-phase method, i.e. minimizing the functional $\mathcal{F}_\alpha$. We report the time required for the whole denoising process and the PSNR of the recovered image. In order to test the speed of the algorithms more fairly, the experiments are repeated 10 times and the average of the 10 timings is given in the tables. The stopping criteria of the minimization phase is set

$$\frac{\|\mathbf{u}^k - \mathbf{u}^{k-1}\|}{\|\mathbf{u}^k\|} \leq 10^{-4} \quad \text{and} \quad \frac{|\mathcal{F}_\alpha(\mathbf{u}^k) - \mathcal{F}_\alpha(\mathbf{u}^{k-1})|}{\mathcal{F}_\alpha(\mathbf{u}^k)} \leq 10^{-4}.$$

The potential function is $\varphi_\alpha(t) = \sqrt{t^2 + \alpha}$ with $\alpha = 100$.

For the conjugate gradient type method, we choose Q_k in (14) to be the identity matrix. To choose μ in Assumption 1, we must have $\mu \geq \max_{\mathbf{v} \in \bar{\Omega}} \|\nabla^2 \mathcal{F}_\alpha(\mathbf{v})\|$. By (15) and the fact that $\nabla^2 \mathcal{F}_\alpha(\mathbf{v})$ is symmetric, we have

$$\|\nabla^2 \mathcal{F}_\alpha(\mathbf{v})\| \leq \|\nabla^2 \mathcal{F}_\alpha(\mathbf{v})\|_\infty \leq 16 \sup_t \varphi_\alpha''(t), \quad \forall \mathbf{v} \in \bar{\Omega}.$$

Therefore, we choose

$$\mu = 16 \sup_t \varphi_\alpha''(t) = \frac{16}{\sqrt{\alpha}},$$

and hence δ in (14) is chosen as

$$\delta = \frac{\sqrt{\alpha-1}}{16} = \frac{\sqrt{99}}{16} < \frac{1}{\mu} = \frac{\sqrt{\alpha}}{16} = \frac{5}{8}.$$

In Table 1, we compare the five nonlinear CG type methods defined in (7)–(11), which are denoted by FR, PR, HS, CD and DY respectively. We see that PR is the most efficient one among the five methods. Therefore, we take PR as a representative of the CG type methods in the following tests. Next, we show the advantages of PR method over the 1D relaxation method applied to the functional (1) as discussed in [4]. The results are given in Table 2. One sees from Table 2 that the CG type method is faster than the relaxation

method when the noise ratio is larger than 50% for both test images. When the noise ratio is 90%, the CG method is about three times faster than the relaxation-based method, i.e. about 60%–70% saving in CPU time. Moreover, we note that the PSNR values attained by the minimizers of (1) and (4) are almost exactly the same.

Table 1. Comparison of the conjugate gradient type methods for *goldhill* image

Noise Ratio	Time					PSNR
	FR	PR	HS	CD	DY	
30%	39.0	28.9	30.6	29.8	44.7	36.0
50%	58.4	43.4	44.8	44.3	67.4	32.7
70%	77.7	59.3	60.3	60.4	90.5	29.8
90%	184	152	153	153	217	26.1

Table 2. Comparison of the conjugate gradient type method with the relaxation-based method

Noise Ratio	<i>goldhill</i>				<i>lena</i>			
	Relaxation		PR		Relaxation		PR	
	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR
30%	35.5	36.0	28.9	36.0	35.7	36.4	49.2	36.5
50%	71.7	32.7	43.4	32.7	85.4	32.9	78.3	33.0
70%	130	29.8	59.3	29.8	133	29.7	81.1	29.8
90%	453	26.1	152	26.1	500	25.3	185	25.4

Finally, Figures 1 and 2 show the results obtained by (i) the adaptive median filter (AMF), (ii) the two-phase schemes solved by 1D relaxation [4], and (iii) the two-phase schemes solved by the conjugate gradient method.

6 Conclusion

In this paper, we give an efficient CG algorithm to minimize the regularization functional in the two-phase impulse removal proposed in [4]. In its original form, the regularization functional is not differentiable because of its non-smooth data-fitting term. We modify it by removing the data-fitting term. Then an efficient CG method, where the line search rule is replaced by a pre-determined step length strategy, is applied to minimize the new functional. Based on the results in [18], global convergence of the algorithm is established. This variant of the two-phase method gives an output having the same visual

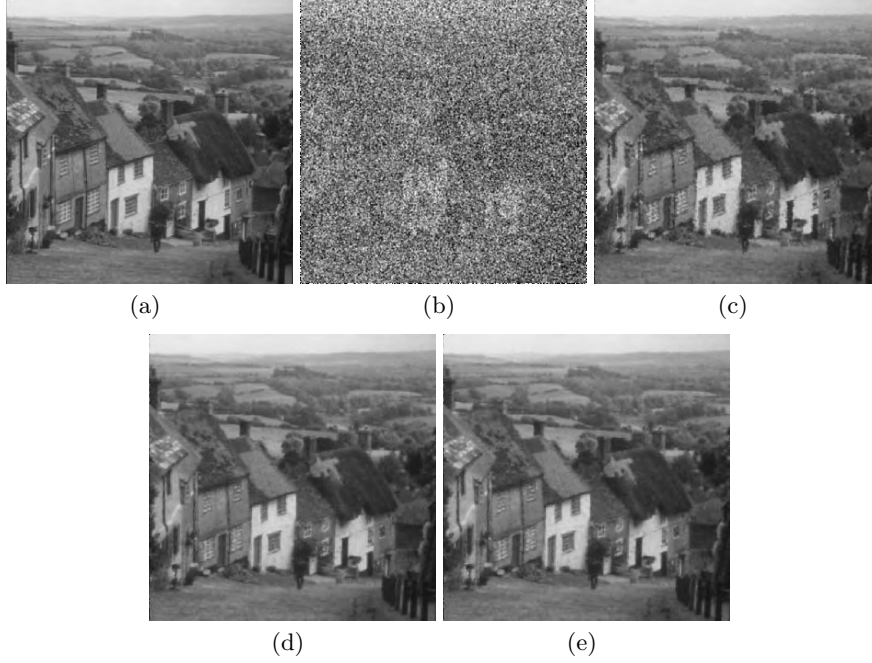


Fig. 1. Restoration results of different algorithms: (a) Original *Goldhill* image, (b) Corrupted *Goldhill* image with 70% salt-and-pepper noise (6.9 dB), (c) Adaptive median filter (26.1 dB), (d) Two-phase method with relaxation (29.8 dB), and (e) Two-phase method with conjugate gradient using (8) for β_k (29.8 dB).

quality as the original method. With slight modification, the CG algorithm can also be applied equally well to random-valued impulse noise (cf. [5]).

Regarding future research directions, we note that in the CG algorithm we are allowed to select a sequence of $\{Q_k\}$ (see (14)) and they are chosen to be the identity in our computations. It would be interesting to define $\{Q_k\}$ according to the Hessian of the objective functional, or further, to perform some preconditioning for the CG algorithm. Preconditioning is not straightforward as the Hessian does not have any special structure. Also here the second order derivative of $\varphi_\alpha(t)$ is only required in the convergence analysis and not in the computation. One may hope to relax the twice continuously differentiable assumption on $\varphi_\alpha(t)$ to only continuously differentiable. This may extend the method to more potential functions such as $\varphi_\alpha(t) = |t|^{1+\epsilon}$, $\epsilon > 0$, which is known to produce better restored images.

References

1. D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

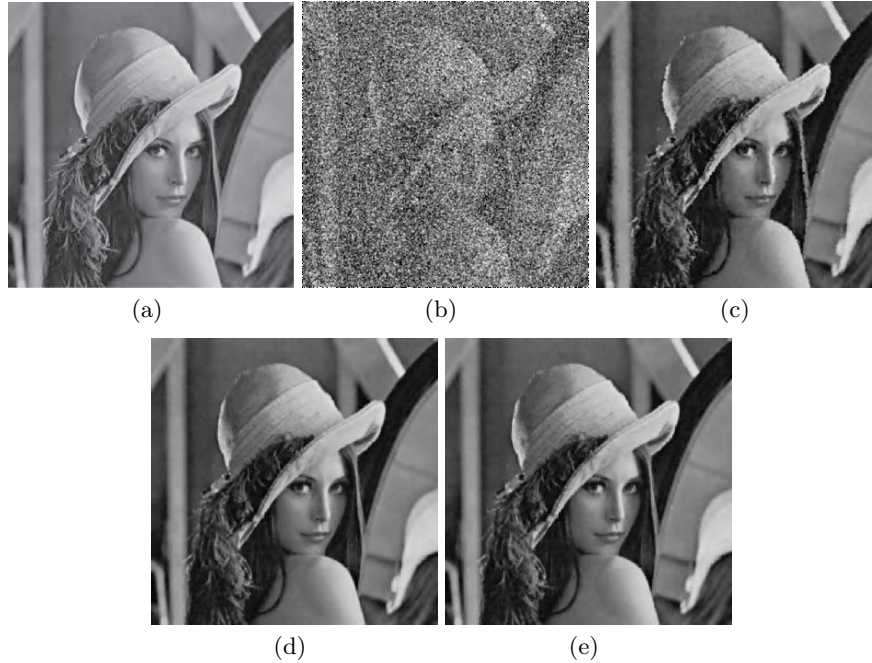


Fig. 2. Restoration results of different algorithms: (a) Original *Lena* image, (b) Corrupted *Lena* image with 70% salt-and-pepper noise (6.7 dB), (c) Adaptive median filter (25.8 dB), (d) Two-phase method with relaxation (29.7 dB), and (e) Two-phase method with conjugate gradient using (8) for β_k (29.8 dB).

2. A. Bovik. *Handbook of Image and Video Processing*. Academic Press, 2000.
3. R. H. Chan, C.-W. Ho, and M. Nikolova. Convergence of Newton's method for a minimization problem in impulse noise removal. *J. Comput. Math.*, 22(2):168–177, 2004.
4. R. H. Chan, C.-W. Ho, and M. Nikolova. Salt-and-pepper noise removal by median-type noise detector and edge-preserving regularization. *IEEE Trans. Image Process.*, 14(10):1479–1485, 2005.
5. R. H. Chan, C. Hu, and M. Nikolova. An iterative procedure for removing random-valued impulse noise. *IEEE Signal Proc. Letters*, 11(12):921–924, 2004.
6. P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Trans. Image Process.*, 6(2):298–311, 1997.
7. T. Chen and H. R. Wu. Adaptive impulse detection using center-weighted median filters. *IEEE Signal Proc. Letters*, 8(1):1–3, 2001.
8. Y. H. Dai and Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.*, 10(1):177–182, 1999.
9. R. Fletcher. *Practical methods of optimization*. A Wiley-Interscience Publication. John Wiley & Sons Ltd., Chichester, second edition, 1987.
10. R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *Comput. J.*, 7:149–154, 1964.

11. P. J. Green. Bayesian reconstructions from emission tomography data using a modified EM algorithm. *IEEE Trans. Medical Imaging*, 9(1):84–93, 1990.
12. M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Research Nat. Bur. Standards*, 49:409–436 (1953), 1952.
13. H. Hwang and R. A. Haddad. Adaptive median filters: new algorithms and results. *IEEE Trans. Image Process.*, 4(4):499–502, 1995.
14. M. Nikolova. A variational approach to remove outliers and impulse noise. *J. Math. Imaging Vision*, 20(1-2):99–120, 2004. Special issue on mathematics and image analysis.
15. E. Polak and G. Ribière. Note sur la convergence de méthodes de directions conjuguées. *Rev. Française Informat. Recherche Opérationnelle*, 3(16):35–43, 1969.
16. W. Rudin. *Principles of mathematical analysis*. McGraw-Hill Book Co., New York, third edition, 1976. International Series in Pure and Applied Mathematics.
17. G. W. Stewart and Ji Guang Sun. *Matrix perturbation theory*. Computer Science and Scientific Computing. Academic Press Inc., Boston, MA, 1990.
18. J. Sun and J. Zhang. Global convergence of conjugate gradient methods without line search. *Ann. Oper. Res.*, 103:161–173, 2001.
19. R. S. Varga. *Matrix iterative analysis*, volume 27 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, expanded edition, 2000.

A Newton-type Total Variation Diminishing Flow

Wolfgang Ring

Institute of Mathematics and Scientific Computing, Faculty of Natural Sciences,
University of Graz. E-mail: wolfgang.ring@uni-graz.at

Summary. A new type of geometric flow is derived from variational principles as a steepest descent flow for the total variation functional with respect to a variable, Newton-like metric. The resulting flow is described by a coupled, non-linear system of differential equations. Written as one scalar evolution equation, the flow equation is non-local. Geometric properties of the flow are investigated, the relation to inverse scale space methods is discussed, and the question of appropriate boundary conditions is addressed. Numerical studies based on a finite element discretization are presented.

Key words: Geometric flow, Newton-type algorithm, mean curvature flow, image processing, inverse scale space

1 Introduction

1.1 Geometric Descent Flow

Geometric flow equations play an important role in image processing especially in image and surface smoothing and feature enhancement. In the image processing context, a gray-scale image is frequently represented by a function $\phi : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$, with $n = 2, 3$ which assigns each point in the image domain U a scalar value between 0 and 1, the gray value of the image at this point. A time-dependent evolution equation of the form

$$\phi_t = A(\phi); \quad \phi_{t=0} = \phi_0 \tag{1}$$

with an appropriate (possibly non-linear) operator A can be used to process a given initial image ϕ_0 . The image ϕ_0 then acts as the initial value for a continuous family of images $\phi(\cdot, t) : U \rightarrow \mathbb{R}$ satisfying (1) for $t \in [0, T]$. It is usually the goal to define the flow equation in such a way that certain desirable features emerge in $\phi(\cdot, t)$ with passing evolution time. Various flows have been designed to reduce noise, to strengthen existing edges in the initial image, or

to identify objects of a certain scale. Of special importance is the extraction or enhancement of *geometrical* features, i.e., properties which depend on the geometry of level-sets of the image function. Equation (1) is said to define a *geometric flow* if the family of level-sets

$$\Gamma_z(t) = \{\mathbf{x} \in U : \phi(\mathbf{x}, t) = z\}$$

for fixed $z \in \mathbb{R}$ depend on the initial data ϕ_0 only via its z -level-set $\Gamma_z(0) = \{\mathbf{x} \in U : \phi_0(\mathbf{x}) = z\}$.

Examples of geometrical flows include mean curvature flow, total variation flow, and Willmore flow along with their various (anisotropic, scale-invariant, multi-channel etc.) versions. See e.g. [17, 24, 12], [28, 31], and [15] and the references therein for more information. All these examples are geometric flows which can be derived from *variational principles* of the form

$$\min_{\phi} \int_{z \in \mathbb{R}} \int_{\Gamma_z} f_z(\mathbf{x}) dS(\mathbf{x}) dz \quad (2)$$

where Γ_z is the z -level-set of ϕ , dS denotes the surface measure on Γ_z and $f_z : \Gamma_z \rightarrow \mathbb{R}$ is a function which depends only on intrinsic (geometrical) properties of Γ_z . If we choose $f_z = 1$, mean curvature flow and total variation flow can be derived as different descent flows for the cost functional (2). (See Subsection 1.2). For the choice $f_z = \kappa^2$, where κ is the mean curvature of the level-set Γ_z , Willmore flow is obtained. Using the co-area formula [16, Thm. 3.4.2] the functional (2) can be rewritten as

$$\min_{\phi} J(\phi) \quad \text{with } J(\phi) = \int_U f(\mathbf{x}, \phi) |\nabla \phi| d\mathbf{x} \quad (3)$$

and $f(\mathbf{x}, \phi) = f_z(\mathbf{x})$ for $\mathbf{x} \in \Gamma_z$. In this reformulation, the dependence of f_z on geometrical properties of Γ_z must be rewritten in terms of differential operators acting on the level-set function ϕ . For (3) standard variational techniques can be applied to obtain a descent flow of the form (1).

Usually, finding the global minimum of (3) is not interesting (frequently the global minimum is attained for $\phi = 0$). One is rather interested in the transition from the initial image function ϕ_0 to an advanced (steady) state ϕ_∞ . In practise, the evolution of ϕ is stopped at an appropriate time T for which the desired image properties are obtained. The descent flow is not uniquely determined by the cost functional (3). Assume, we have chosen a Banach space X with $\phi \in X$ and we denote its dual space by X' . The usual procedure is to differentiate the functional J with respect to ϕ to get a first order approximation of the form

$$J'(\phi)(\psi) = \langle F, \psi \rangle_{X', X}$$

and to choose a descent direction $\psi_{\text{desc}} \in X$ as the dual element to the negative derivative $-F \in X'$. A dual element to $-F$ is defined by the properties $\langle F, \psi \rangle_{X', X} = -\|F\|_{X'}$ and $\|\psi\|_X = 1$. It is uniquely determined by F if the

space X is reflexive and strictly convex [3, Thm. 1.2]. Obviously, the choice of ψ_{desc} does depend not only on F but also on the chosen function space X , especially on its norm $\|\cdot\|_X$ and the duality pairing $\langle \cdot, \cdot \rangle_{X', X}$. Note that equivalent norms do not yield identical dual elements.

Most constructions of descent flows for geometric functionals use a variant of an L^2 -type norm to establish a connection between derivative and descent direction. In this paper we define and investigate an alternative norm for the construction of a descent direction with respect to the functional (2) for the special case $f_z = 1$. This norm is defined using a positive definite approximation of the Hessian of the cost functional. This puts our approach within the class of *Newton-type techniques*. Moreover, the norm depends on the current level-set function ϕ , thus we have a *variable norm* (variable metric) with every $t \in [0, T]$. Doing so, we mimic Newton's method from nonlinear optimization which aims at finding a descent direction pointing towards the minimum of a second-order approximation of the cost functional at every step of the algorithm (see [26] for details). It turns out that the operator on the right-hand side of the resulting flow equation (1) is non-local and the speed of propagation of level-sets depends on global properties such as their overall surface measure. The flow has certain similarities with the recently suggested inverse scale space approach for image smoothing. In the following, we present a derivation of the flow equation and a discussion of some of its geometric properties. We give numerical examples to investigate the behavior of the flow on individual level-sets and on the collection of all level-sets of a given image. We evaluate the dependence of the flow on certain parameters in the model and show that our approach is capable of interpolating between mean curvature flow and shrinking of shapes with constant speed.

1.2 Mean Curvature and Bounded Variation Flows

We choose the following notational convention: points in $\mathbb{R}^n$ as well as (tangential) vectors will be denoted by boldface letters e.g., $\mathbf{x}$ and $\mathbf{v}$. A vector $\mathbf{v}$ is always a (contravariant) column vector

$$\mathbf{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}.$$

The corresponding (covariant) row vector is denoted by $\mathbf{v}^t$. The gradient ∇f of a function $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$ is always a row vector. The scalar product of vectors $\mathbf{v}$ and $\mathbf{w}$ in $\mathbb{R}^n$ is denoted by $\langle \mathbf{v}, \mathbf{w} \rangle$ and sometimes also by $\mathbf{v}^t \cdot \mathbf{w}$ if this notation makes the presentation more transparent. We write the tensor product of two vectors $\mathbf{v}$ and $\mathbf{w}$ as $\mathbf{v} \otimes \mathbf{w} = \mathbf{v} \cdot \mathbf{w}^t = (v_i w_j)_{i,j=1}^n$.

The starting point for our considerations is the geometric surface area functional

$$J(\Gamma) = \int_{\Gamma} 1 \, dS = |\Gamma| \quad (4)$$

where $\Gamma \subset \mathbb{R}^n$ is an $(n-1)$ -dimensional subset (i.e., its $(n-1)$ -dimensional Hausdorff measure is finite) and dS denotes the integration with respect to the $(n-1)$ -dimensional Hausdorff measure on Γ . We are specifically interested in the situation where $\Gamma = \Gamma_z$ is the z -level set of a function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$, i.e.,

$$\Gamma_z = \{\mathbf{x} \in \mathbb{R}^n : \phi(\mathbf{x}) = z\} \quad (5)$$

with some constant $z \in \mathbb{R}$. Usually we assume that $\Gamma = \partial\Omega$ with a bounded open set $\Omega \subset \mathbb{R}^n$. In the context of the level-set formulation (5) we assume that $\Gamma_z = \partial\Omega_z$ with $\Omega_z = \{\mathbf{x} \in \mathbb{R}^n : \phi(\mathbf{x}) < z\}$.

We are interested in *area diminishing flows* i.e., smoothly varying families of surfaces $\{\Gamma(t) : t \geq 0\}$ for which $|\Gamma(t)|$ is decreasing with increasing t . An obvious possibility to construct an area diminishing flow is to calculate the derivative of (4) with respect to Γ and to propagate Γ in a direction for which the directional derivative is negative. It is well known that the derivative of the area functional with respect to the geometric variable Γ is given as

$$dJ(\Gamma; \mathbf{v}) = \int_{\Gamma} \kappa \langle \mathbf{v}, \mathbf{n} \rangle \, dS \quad (6)$$

where $\mathbf{n}$ is the unit exterior normal vector to Γ , κ denotes the mean curvature of Γ and $\mathbf{v} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the direction of perturbation of Γ . See for example [29, p. 116, eq. (2.174)] [13, p. 356, eq. (4.21)], [30] for a parameter free derivation of the result, see e.g., [6], [32] for arguments using parametric curves and [2] for a very instructive comparison of the two approaches. To define a derivative it is necessary to specify the admissible perturbations of the independent variable. In the case of the geometric functional (6) we consider perturbations which are (in first order) of the form

$$\Gamma^h = \{\mathbf{x}^h = \mathbf{x} + h\mathbf{v}(\mathbf{x}) : \mathbf{x} \in \Gamma\}.$$

With this (6) is obtained as the directional derivative

$$dJ(\Gamma; \mathbf{v}) = \lim_{h \rightarrow 0} \frac{1}{h} (J(\Gamma^h) - J(\Gamma)).$$

Suppose we choose $\mathbf{v} = -\kappa\mathbf{n}$ as direction of propagation. Of course, this choice is motivated by the fact that the directional derivative (6) is always non-positive in this direction. The corresponding area diminishing flow is the solution to the partial differential equation

$$\frac{\partial \Gamma}{\partial t} = -\kappa\mathbf{n}. \quad (7)$$

Equation (7) is called the geometric heat equation and is extensively investigated in the literature (see [5] and the numerous references cited therein).

Using Osher's and Sethian's idea [27], the propagation of the surface $\Gamma(t)$ can equivalently be expressed by a propagation law for a (now time-dependent) level set function $\phi(t, \mathbf{x})$. In this context the family of surfaces $\Gamma(t)$ is implicitly given as $\Gamma(t) = \Gamma_0(t) = \{\mathbf{x} \in \mathbb{R}^n : \phi(t, \mathbf{x}) = 0\}$. The propagation of $\Gamma(t)$ with velocity given by $\mathbf{v} = F\mathbf{n}$ is translated into a propagation law for the level-set function:

$$\phi_t + F|\nabla\phi| = 0. \quad (8)$$

Setting $F = -\kappa$ in the level set equation (8) and noting that normal vector and curvature can be expressed as

$$\mathbf{n}^t = \frac{\nabla\phi}{|\nabla\phi|} \text{ and } \kappa = \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \quad (9)$$

the *mean curvature flow equation*

$$\phi_t = |\nabla\phi|\operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \quad (10)$$

is obtained. The zero-level set of ϕ plays no specific role in the implicit formulation (10). In fact, not only the zero-level set but *all level sets* of ϕ evolve according to the geometric heat equation (7). See [17, 18, 19, 20] and [8] for theoretical results and [7, 10, 11] for numerical algorithms.

We present yet another derivation of (10) from a variational principle which is taken (at least in spirit) from [15]. For a Lipschitz continuous level set function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ with compact support we consider the functional

$$\tilde{J}(\phi) = \int_{z \in \mathbb{R}} \int_{\mathbf{x} \in \Gamma_z} 1 \, dS(\mathbf{x}) \, dz = \int_{\mathbf{x} \in \mathbb{R}^n} |\nabla\phi(\mathbf{x})| \, d\mathbf{x}. \quad (11)$$

where we used the co-area formula [16, p. 112, Thm. 3.4.2] for the equivalence of the two expressions above. A sufficient requirement on the function ϕ for the co-area formula to hold is that ϕ is Lipschitz continuous. In the case that $|\nabla\phi|$ is not integrable on $\mathbb{R}^n$, equation (11) reproduces the trivial identity $\infty = \infty$. To avoid this type of degeneracy, we shall assume that ϕ is constant, and hence $\nabla\phi = 0$ outside a ball of sufficiently large radius. Alternatively one can restrict the domain of integration to an arbitrary measurable subset $A \subset \mathbb{R}^n$ and work with the following form of the co-area formula:

$$\int_{z \in \mathbb{R}} \int_{\mathbf{x} \in \Gamma_z \cap A} 1 \, dS(\mathbf{x}) \, dz = \int_{\mathbf{x} \in A} |\nabla\phi(\mathbf{x})| \, d\mathbf{x}. \quad (12)$$

From the left-hand side of (11) it follows that $\tilde{J}$ simultaneously measures the areas of all level sets of the function ϕ in an integrated (averaged) form.

We now construct a flow for the level set function ϕ which diminishes the functional (11). The following calculations are formal i.e., without the

necessary specification of the respective function spaces. Also the singularity for $|\nabla\phi| = 0$ is treated formally and $|\nabla\phi|$ is replaced by $\sqrt{|\nabla\phi|^2 + \epsilon^2}$ in the denominator with small $\epsilon > 0$ for the concrete numerical calculations. Differentiation of $\tilde{J}$ with respect to ϕ gives

$$\tilde{J}'(\phi) \psi = \frac{\partial \tilde{J}}{\partial \phi} \psi = \int_{\mathbb{R}^n} \frac{\langle \nabla \phi, \nabla \psi \rangle}{|\nabla \phi|} d\mathbf{x} \quad (13)$$

for the derivative of $\tilde{J}$ with respect to ϕ in the direction ψ . We can find a direction of propagation ψ_{desc} for an area diminishing flow by minimizing the predicted descent (13) with respect to ψ under the norm constraint

$$\|\psi\|_{\text{desc}}^2 = 1, \quad (14)$$

where $\|\cdot\|_{\text{desc}}$ is an appropriate norm for the descent direction ψ . We want the propagation to be of level-set form i. e.

$$\phi_t - \psi = \phi_t + F|\nabla\phi| = 0, \quad (15)$$

that is, we set $\psi = -F|\nabla\phi|$. This, and the first variant of the cost functional (11) motivates the choice of the norm for ψ in (14) as the L^2 -norm of the equivalent speed function F on Γ_z integrated over all $z \in \mathbb{R}$. More precisely we use $\|\psi\|_{\text{desc}} = \|\psi\|_{L^2, \phi}$ with

$$\begin{aligned} \|\psi\|_{L^2, \phi}^2 &= \int_{z \in \mathbb{R}} \int_{\Gamma_z} F^2(\mathbf{x}) dS(\mathbf{x}) dz = \int_{\mathbf{x} \in \mathbb{R}^n} F^2(\mathbf{x}) |\nabla \phi(\mathbf{x})| d\mathbf{x} \\ &= \int_{\mathbf{x} \in \mathbb{R}^n} \psi^2(\mathbf{x}) |\nabla \phi(\mathbf{x})|^{-1} d\mathbf{x}. \end{aligned} \quad (16)$$

To derive the necessary optimality conditions for the minimization of (13) under the constraint (14) with norm given by (16) we introduce the Lagrange functional

$$\mathcal{L}(\psi, \lambda) = \int_{\mathbb{R}^n} \frac{\langle \nabla \phi, \nabla \psi \rangle}{|\nabla \phi|} d\mathbf{x} + \lambda \left(\int_{\mathbb{R}^n} \psi^2 |\nabla \phi|^{-1} d\mathbf{x} - 1 \right). \quad (17)$$

From the Karush-Kuhn-Tucker condition $\mathcal{L}_\psi = 0$ we get

$$\text{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + 2\lambda \psi |\nabla \phi|^{-1} = 0.$$

Consequently

$$\psi_{\text{desc}} = \frac{1}{2\lambda} |\nabla \phi| \text{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right),$$

where $\lambda > 0$ is chosen such that $\|\psi\|_{L^2, \phi} = 1$. Inserting this in (15) we get (after a re-scaling of the time variable) the mean curvature flow equation (10).

Suppose we had chosen a different norm $\|\cdot\|_{\text{desc}}$ as given in (16). Then the resulting optimality system has a different form and we arrive at a flow equation for ϕ which is different from (10) but still has the property to decrease the functional (11) although along a different trajectory. As a second possibility we choose $\|\cdot\|_{\text{desc}} = \|\cdot\|_{L^2}$ where

$$\|\psi\|_{L^2}^2 = \int_{\mathbb{R}^n} \psi^2 d\mathbf{x}, \quad (18)$$

the usual (non-geometric) L^2 -norm on $\mathbb{R}^n$. With this, the necessary optimality condition reads as

$$\operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + 2\lambda\psi = 0,$$

and hence

$$\psi_{\text{desc}} = \frac{1}{2\lambda} \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right).$$

The corresponding flow equation is the well investigated *total variation flow*

$$\phi_t = \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right). \quad (19)$$

See e.g., [28, 31, 14, 1].

1.3 Outline

Our paper is concerned with the construction of a different class of norms $\|\cdot\|_{\text{desc}}$ for the determination of the descent direction. The norm in the new approach is derived from the second order approximation of the cost functional at the current argument ϕ . Consequently, the norm varies along with the propagating level-set function. This is, by the way, also the case for the geometric L^2 -norm (16). The approach can therefore be considered as variable metric or Newton-type equivalent to the first order, gradient-type flows (10) and (19). In the following we present a derivation of the flow equations, we investigate geometric properties of the flow, we discuss the relation of the approach to inverse scale space methods, and we report the results of our numerical and experimental investigations for the new approach. In Section 2 we construct a norm from the Hessian of the cost functional (11) and we derive the corresponding steepest descent flow. In Section 3, theoretical arguments are presented which shed some light on the geometrical and qualitative behavior of the flow. Moreover, the relation to inverse scale space methods is discussed. Boundary conditions are introduced and an alternative formulation of the system of flow equations is derived. Section 4 contains numerical examples and parameter studies. The behavior of the flow on individual level-sets on the one hand, and simultaneously on all level-sets of an image on the other hand, is investigated.

2 A Newton-type Flow for the Minimization of the Area of Level-Sets

We aim for the construction of an alternative evolution equation for the level set function ϕ which also has the property of decreasing the areas of level sets with increasing time. We do so by replacing the vector field $-\kappa \mathbf{n}$ in (7) by vector fields of the form $F \mathbf{n}$ where the scalar function F is found as a steepest descent direction. The corresponding metrics are constructed from the second order derivatives of the area functionals (4) and (11) respectively.

Formal differentiation of (13) with respect to ϕ gives the second derivative of the functional (11) as the bilinear form

$$\tilde{J}''(\phi)(\psi, \eta) = \int_{\mathbb{R}^n} \frac{1}{|\nabla \phi|} \left(\langle \nabla \psi, \nabla \eta \rangle - \frac{1}{|\nabla \phi|^2} \langle \nabla \phi, \nabla \psi \rangle \langle \nabla \phi, \nabla \eta \rangle \right) d\mathbf{x}. \quad (20)$$

The classical Newton (variable metric) approach suggests to use $\|\psi\|_{\text{desc}}^2 = \tilde{J}''(\phi)(\psi, \psi)$ provided that the bilinear form on the right-hand side is positive definite. Since

$$|\nabla \psi|^2 - \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \psi \right\rangle^2 \geq |\nabla \psi|^2 - \left| \frac{\nabla \phi}{|\nabla \phi|} \right|^2 |\nabla \psi|^2 = 0$$

we find that the bilinear form (20) is positive semi-definite. It is not positive definite since $\tilde{J}''(\phi)(\psi, \psi) = 0$ for any ψ which is constant on level sets of ϕ . In fact, if we set $\psi = f(\phi)$ with an arbitrary (smooth enough) function $f : \mathbb{R} \rightarrow \mathbb{R}$, we obtain

$$|\nabla \psi|^2 - \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \psi \right\rangle^2 = f'(\phi)^2 |\nabla \phi|^2 - (f'(\phi) |\nabla \phi|)^2 = 0.$$

We use a positive definite approximation to the Hessian as metric for the descent direction thus putting our approach into the class of inexact Newton methods. More precisely we set $\|\psi\|_{\text{desc}} = \|\psi\|_{N, \phi}$ (the subscript ‘N’ stands for a Newton-type metric and ϕ indicates the dependence of the variable metric on the current level-set function) with

$$\|\psi\|_{N, \phi}^2 = \tilde{J}''(\phi)(\psi, \psi) + \alpha \|\psi\|_{L^2, \phi}^2 \quad (21)$$

with some parameter $\alpha > 0$ and $\|\psi\|_{L^2, \phi}$ as given in (16). Alternatively, we shall also use

$$\|\psi\|_{N1, \phi}^2 = \tilde{J}''(\phi)(\psi, \psi) + \alpha \|\psi\|_{L^2}^2, \quad (22)$$

with $\|\psi\|_{L^2}$ defined in (18). As descent direction we choose the solution to the constrained minimization problem

$$\begin{aligned} & \min_{\psi} J'(\phi) \psi \\ & \text{such that } \|\psi\|_{N, \phi}^2 = 1. \end{aligned} \quad (23)$$

The Lagrange functional for the constrained optimization problem (23) with the choice (21) for $\|\cdot\|_{N,\phi}$ is given as

$$\begin{aligned} \mathcal{L}_{N,\phi}(\psi, \lambda) = & \int_{\mathbb{R}^n} \frac{\langle \nabla \phi, \nabla \psi \rangle}{|\nabla \phi|} d\mathbf{x} + \\ & \lambda \left(\int_{\mathbb{R}^n} \frac{1}{|\nabla \phi|} \left(|\nabla \psi|^2 - \frac{1}{|\nabla \phi|^2} \langle \nabla \phi, \nabla \psi \rangle^2 \right) d\mathbf{x} + \alpha \int_{\mathbb{R}^n} \frac{\psi^2}{|\nabla \phi|} d\mathbf{x} - 1 \right). \end{aligned} \quad (24)$$

The necessary optimality condition $\partial_\psi \mathcal{L}_{N,\phi} = 0$ yields that the minimizer $\hat{\psi}$ to (23) and the optimal multiplier $\hat{\lambda}$ must satisfy

$$\begin{aligned} \int_{\mathbb{R}^n} \frac{1}{|\nabla \phi|} \left(\langle \nabla \hat{\psi}, \nabla \eta \rangle - \frac{1}{|\nabla \phi|^2} \langle \nabla \phi, \nabla \hat{\psi} \rangle \langle \nabla \phi, \nabla \eta \rangle \right) d\mathbf{x} + \alpha \int_{\mathbb{R}^n} \frac{\hat{\psi} \eta}{|\nabla \phi|} d\mathbf{x} \\ = -\frac{1}{2\hat{\lambda}} \int_{\mathbb{R}^n} \frac{\langle \nabla \phi, \nabla \eta \rangle}{|\nabla \phi|} d\mathbf{x} \end{aligned} \quad (25)$$

for all test functions $\eta \in \mathcal{D}(\mathbb{R}^n)$. If we had used the norm (22) instead of (21), the last term in (25) must be replaced by $\alpha \int_{\mathbb{R}^n} \hat{\psi} \eta d\mathbf{x}$. Again, the Lagrange multiplier can be chosen $\hat{\lambda} = \frac{1}{2}$ since any other choice of the multiplier only modifies the size of the descent direction $\hat{\psi}$ and any such modification can be compensated by a re-scaling of the time variable in (15).

We have, therefore, found a weak form of a second order (Newton type) descent flow for the cost functional (11) with respect to the metric (21) as the coupled system

$$\begin{aligned} \int_{\mathbb{R}^n} \frac{1}{|\nabla \phi|} \left(\langle \nabla \psi, \nabla \eta \rangle - \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \psi \right\rangle \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \eta \right\rangle + \alpha \psi \eta \right) d\mathbf{x} \\ = - \int_{\mathbb{R}^n} \frac{\langle \nabla \phi, \nabla \eta \rangle}{|\nabla \phi|} d\mathbf{x}, \end{aligned} \quad (26a)$$

$$\phi_t = \psi \quad \text{on } \mathbb{R}^n, \quad (26b)$$

for all test functions $\eta \in \mathcal{D}(\mathbb{R}^n)$. Here (and in the following) we have changed the notation from the specific solution $\hat{\psi}$ of (23) back to the generic ψ .

Next, we rewrite equation (26a) in strong form. Beforehand, we provide a few useful relations. With (9) we get for the derivative of the mapping $\mathbf{n} : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$$D\mathbf{n} = \frac{1}{|\nabla \phi|} (D^2 \phi - \mathbf{n} \cdot \mathbf{n}^t \cdot D^2 \phi) = \frac{1}{|\nabla \phi|} (I - \mathbf{n} \otimes \mathbf{n}) \cdot D^2 \phi. \quad (27)$$

at all points where $\mathbf{n}$ is differentiable. Note that $D\mathbf{n}(\mathbf{x}) \in \mathbb{R}^{3 \times 3}$ in (27) is *not* the intrinsic shape operator (the Weingarten map) $D\mathbf{n} : T_{\mathbf{x}}\Gamma \rightarrow T_{\mathbf{x}}\Gamma$ as e.g., described in [25]. Especially, $D\mathbf{n}$ in our case is not symmetric, the asymmetry being related to the behavior of $D\mathbf{n}$ in normal direction. For later use, we make the following consideration. From $\mathbf{n}^t \cdot \mathbf{n} = 1$ we get by differentiation

$$0 = \nabla(\mathbf{n}^t \cdot \mathbf{n}) = 2 \mathbf{n}^t \cdot D\mathbf{n}. \quad (28)$$

Note that, on the other hand, $D\mathbf{n} \cdot \mathbf{n} \neq 0$ for each (generic) situation where the normals to level sets of ϕ are not straight lines. This implies that $D\mathbf{n}$ is not symmetric in general. We shall also use the relation

$$\nabla\left(\frac{1}{|\nabla\phi|}\right) = -\frac{1}{|\nabla\phi|^2} \mathbf{n}^t \cdot D^2\phi. \quad (29)$$

To find a strong formulation for equation (25) we use Green's formula on all terms $\nabla\eta$ in (25). We obtain

$$-\operatorname{div}\left(\frac{1}{|\nabla\phi|}\left(\nabla\psi - \left\langle \frac{\nabla\phi}{|\nabla\phi|}, \nabla\psi \right\rangle \frac{\nabla\phi}{|\nabla\phi|}\right)\right) + \alpha \frac{\psi}{|\nabla\phi|} = \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right). \quad (30)$$

We introduce the Hessian operator $H(\phi)$ as

$$H(\phi) \cdot \psi = -\operatorname{div}\left(\frac{1}{|\nabla\phi|}\left(\nabla\psi - \left\langle \frac{\nabla\phi}{|\nabla\phi|}, \nabla\psi \right\rangle \frac{\nabla\phi}{|\nabla\phi|}\right)\right) + \alpha \frac{\psi}{|\nabla\phi|}. \quad (31)$$

With this, (26) can be (formally) written as an evolution equation for ϕ :

$$\phi_t = (H(\phi))^{-1} \left[\operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \right]. \quad (32)$$

For fixed $\phi \in \mathcal{C}^1(\mathbb{R}^n)$ and if $|\nabla\phi|$ in the denominators is replaced by a strictly positive approximation, equation (30) is a proper, degenerate elliptic equation in the sense of viscosity solutions. It can be shown, using Perrons method for existence and a comparison principle for uniqueness that (30) has a unique, locally Lipschitz continuous viscosity solution ψ . See [9] for a comprehensive introduction to the theory of viscosity solutions for degenerate elliptic equation. So far, we can only proof existence and uniqueness for equation (30) but not for the coupled system (30) and (26b) (or for the integrated formulation (32)). A thorough theoretical investigation of (32) is going to be the content of future work. Note that the evolution equation (32) is *non-local* in the spacial variable $\mathbf{x}$ since H^{-1} is a non-local operator.

We write (30) in a more geometric form. Resolving the div-operator and using the notation (9) yields for the first term on the left hand side of (30):

$$\begin{aligned}
& \frac{1}{|\nabla\phi|^2} \left\langle \mathbf{n}^t \cdot D^2\phi, \nabla\psi - \langle \mathbf{n}, \nabla\psi \rangle \mathbf{n}^t \right\rangle \\
& \quad - \frac{1}{|\nabla\phi|} \left(\Delta\psi - \mathbf{n}^t \cdot D^2\psi \cdot \mathbf{n} - \nabla\psi \cdot D\mathbf{n} \cdot \mathbf{n} - (\nabla\psi \cdot \mathbf{n}) \operatorname{div}(\mathbf{n}) \right) \\
& = \frac{1}{|\nabla\phi|^2} \left(\nabla\psi \cdot (I - \mathbf{n} \otimes \mathbf{n}) \cdot D^2\phi \cdot \mathbf{n} \right) \\
& \quad - \frac{1}{|\nabla\phi|} \left(\Delta\psi - \mathbf{n}^t \cdot D^2\psi \cdot \mathbf{n} - (\nabla\psi \cdot \mathbf{n}) \operatorname{div}(\mathbf{n}) - \right. \\
& \quad \quad \left. \frac{1}{|\nabla\phi|} (\nabla\psi \cdot (I - \mathbf{n} \otimes \mathbf{n}) \cdot D^2\phi \cdot \mathbf{n}) \right) \\
& = \frac{2}{|\nabla\phi|^2} \left(\nabla\psi \cdot (I - \mathbf{n} \otimes \mathbf{n}) \cdot D^2\phi \cdot \mathbf{n} \right) \\
& \quad - \frac{1}{|\nabla\phi|} \left(\Delta\psi - \mathbf{n}^t \cdot D^2\psi \cdot \mathbf{n} - (\nabla\psi \cdot \mathbf{n}) \operatorname{div}(\mathbf{n}) \right).
\end{aligned}$$

Here we used the identities (29) and (27). With this, we obtain

$$\frac{1}{|\nabla\phi|} \left(-(\Delta\psi - \mathbf{n}^t \cdot D^2\psi \cdot \mathbf{n} - (\nabla\psi \cdot \mathbf{n}) \operatorname{div}(\mathbf{n})) + 2\nabla\psi \cdot D\mathbf{n} \cdot \mathbf{n} + \alpha\psi \right) = \operatorname{div}(\mathbf{n}) \quad (33)$$

for an equivalent formulation to (30).

Yet another — in certain respects more compact — variant of the system (33) can be found if we use the scalar speed function F as update direction instead of ψ . Setting $\psi = F|\nabla\phi|$ and using

$$\frac{\nabla\psi}{|\nabla\phi|} = \nabla F + \frac{F}{|\nabla\phi|^2} \nabla\phi \cdot D^2\phi$$

we obtain for (30)

$$\begin{aligned}
& -\operatorname{div} \left(\nabla F + \frac{F}{|\nabla\phi|^2} \nabla\phi \cdot D^2\phi - \left\langle \mathbf{n}, \nabla F + \frac{F}{|\nabla\phi|^2} \nabla\phi \cdot D^2\phi \right\rangle \mathbf{n}^t \right) + \alpha F \\
& = -\operatorname{div} \left(\nabla F - \langle \nabla F, \mathbf{n} \rangle \mathbf{n}^t + \frac{F}{|\nabla\phi|} \mathbf{n}^t \cdot D^2\phi \cdot (I - \mathbf{n} \otimes \mathbf{n}) \right) + \alpha F \\
& = -\operatorname{div} (\nabla F - \langle \nabla F, \mathbf{n} \rangle \mathbf{n}^t + F (D\mathbf{n} \cdot \mathbf{n})^t) + \alpha F \\
& = -(\Delta F - \mathbf{n}^t \cdot D^2 F \cdot \mathbf{n} - \operatorname{div}(\mathbf{n}) \nabla F \cdot \mathbf{n}) + F (\alpha + \operatorname{div}(D\mathbf{n} \cdot \mathbf{n})) = \operatorname{div}(\mathbf{n}). \tag{34}
\end{aligned}$$

As an equation for F the formulation (34) is somewhat less complicated than equation (33). However, the term $\operatorname{div}(D\mathbf{n} \cdot \mathbf{n})$ comprises third derivatives of ϕ and the coupled system (34) together with the level set equation $\phi_t + F|\nabla\phi| = 0$ is more involved. We mostly deal with the system (33) in the subsequent considerations.

3 Geometric Properties

In this section we investigate geometric and qualitative properties of the flow (33). We will show that the flow is close to intrinsic in the sense that the speed ψ depends only on geometric properties of the level-sets of the function ϕ , on $D\mathbf{n}$ and on $|\nabla\phi|$. We also consider the similarities and differences to a flow with constant normal speed and to inverse scale space methods.

3.1 Intrinsic Formulation

We make a few comments on the terms in (33). Using the differential geometric notation (9) we get

$$\Delta\psi - \mathbf{n}^t \cdot D^2\psi \cdot \mathbf{n} - (\nabla\psi \cdot \mathbf{n}) \operatorname{div}(\mathbf{n}) = \Delta\psi - \frac{\partial^2\psi}{\partial n^2} - \kappa \frac{\partial\psi}{\partial n} = \Delta_\Gamma\psi$$

where Δ_Γ denotes the *Laplace-Beltrami operator* with respect to the level sets of ϕ (see [29, Prop. 2.68, p. 94]). The differential operator Δ_Γ is intrinsic on level sets of ϕ , i.e., $\Delta_\Gamma\psi(\mathbf{x})$ depends only on values of ψ on the level set $\{\mathbf{y} : \phi(\mathbf{y}) = \phi(\mathbf{x})\}$.

The term in $\nabla\psi \cdot D\mathbf{n} \cdot \mathbf{n}$ (33) is in fact the directional derivative of ψ with direction given by $D\mathbf{n} \cdot \mathbf{n}$. Due to (28) we have $\mathbf{n}^t \cdot D\mathbf{n} \cdot \mathbf{n} = 0$ hence, the normal component of the direction vanishes, i.e., $\nabla\psi \cdot D\mathbf{n} \cdot \mathbf{n}$ is a *tangential derivative* with respect to the level set $\{\phi = \text{const}\}$. It therefore follows that the differential equation (33) is intrinsic with on level sets of ϕ and can (or could if we wanted to resolve level sets) be solved independently for ψ on each level set of ϕ . The corresponding form of (33) is given by

$$-\Delta_\Gamma\psi + 2\nabla\psi \cdot D\mathbf{n} \cdot \mathbf{n} + \alpha\psi = |\nabla\phi| \kappa \quad \text{on } \Gamma \quad (35)$$

for each level-set $\Gamma = \Gamma_z = \{\phi = z\}$. Equation (35) is intrinsic for the variable ψ . The coefficients of (35), however, depend not only on the geometry of level sets. The terms $D\mathbf{n} \cdot \mathbf{n}$ on the left-hand side and $|\nabla\phi|$ on the right-hand side are not only on the geometry of the level-sets, but constitute a coupling between level-sets.

The nature of the non-intrinsic terms becomes more apparent in the formulation (34). We have

$$\operatorname{div}(D\mathbf{n} \cdot \mathbf{n}) = \operatorname{tr}(D\mathbf{n} \cdot D\mathbf{n}^t) + \langle \nabla \operatorname{div}(\mathbf{n}), \mathbf{n} \rangle = \sum_{i=1}^{n-1} \kappa_i^2 + \partial_{\mathbf{n}}\kappa,$$

where κ_i denote the principal curvatures of Γ . With this, we arrive at the system

$$-\Delta_\Gamma F + \left(\alpha + \sum_{i=1}^{n-1} \kappa_i^2 + \partial_{\mathbf{n}}\kappa\right) F = \kappa, \quad (36a)$$

$$\phi_t + F |\nabla\phi| = 0. \quad (36b)$$

The level-set equation (36b) propagates each individual level-set of ϕ with speed given by F independently of the other level-sets. Thus, the directional derivative $\partial_{\mathbf{n}}\kappa$ constitutes the only coupling between level-sets in the system (36).

3.2 Relation to Flow with Constant Speed

To obtain more inside in the qualitative behavior of the system (26) we consider the special situation $\phi = b_\Gamma$ where b_Γ is the signed distance function to the set $\Gamma = \partial\Omega$ with a smooth bounded open set Ω . In this case, we have $\mathbf{n}^t = \nabla b_\Gamma$, $\kappa = \Delta b_\Gamma$, $D\mathbf{n} = D^2 b_\Gamma$, $D\mathbf{n} \cdot \mathbf{n} = (\mathbf{n}^t \cdot D\mathbf{n})^t = 0$, and $|\nabla\phi| = |\nabla b_\Gamma| = 1$. Thus, equation (35) reduced to

$$-\Delta_\Gamma \psi + \alpha \psi = \kappa. \quad (37)$$

on Γ .

Specifically, for the 2-dimensional situation $n = 2$, we consider the closed level curve Γ which we write in parametrized form $\Gamma = \{\mathbf{x}(s) : s \in [0, L]\}$ with s denoting arc-length and $\mathbf{x}(0) = \mathbf{x}(L)$. Let us set $\alpha = 0$ for the moment. Equation (37) can then be written as

$$-\psi''(s) = \kappa(s)$$

for $s \in [0, L]$ where we set $\psi(s) = \psi(\mathbf{x}(s))$. Integrating with respect to s gives

$$-\int_0^L \psi''(s) ds = -\psi'(L) + \psi'(0) = 0$$

for the left-hand side whereas the right-hand side gives

$$\int_0^L \kappa(s) ds = \int_0^L \theta'(s) ds = 2\pi$$

where θ is the angle between the tangential direction along Γ and a fixed reference direction. Therefore, (37) is not solvable for $\alpha = 0$. For different choices $\alpha > 0$ the solution to (37) with an oscillating right-hand side is shown in Figure 1. The first graphic shows the curvature κ along the curve Γ . The remaining five graphs show the solution to $-\psi'' + \alpha\psi = \kappa$ for $\alpha = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$. We plotted the graphs for the different α -s of different scales. More precisely, the graphics are scaled such that each ψ is shown in the range between 80% and 130% of the average function value. Using the same scale for all graphics would not be useful since the magnitude of the ϕ grows approximately like $1/\alpha$. A variable scaling, however, allows to compare the amount of relative variation, i.e. variation divided by the average, in ψ for different α values. Since ϕ acts as an update direction and scaling in ϕ can always be compensated by the corresponding reciprocal scaling in the

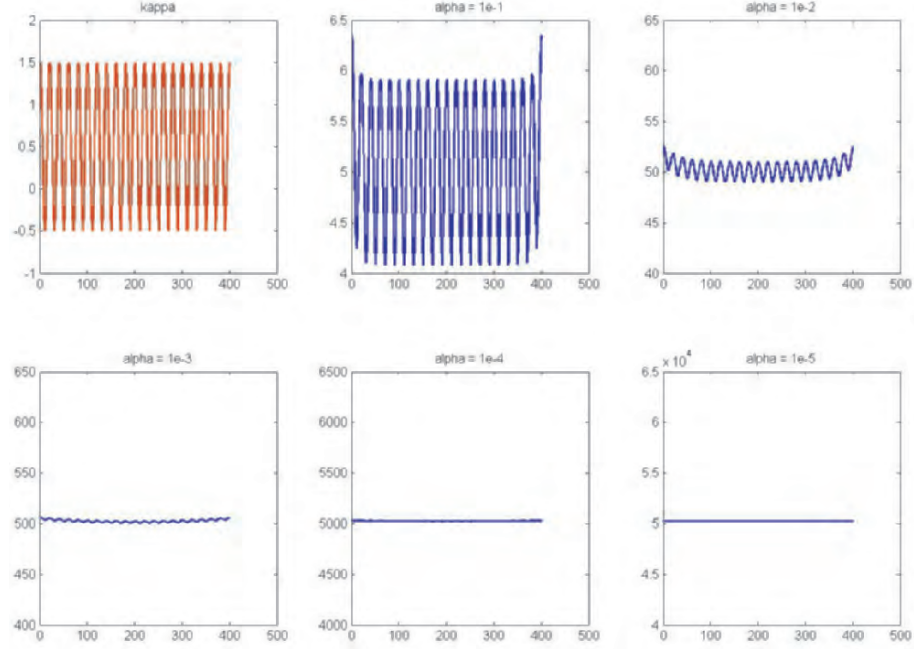


Fig. 1. Behavior of the solution to $-\psi'' + \alpha\psi = \kappa$ for vanishing α with oscillating right-hand side.

size of the time-step, the graphical representation in Figure 1 is appropriate. It can be seen that ψ becomes approximately constant for small α in the sense that the oscillations become insignificant compared to the overall value of ψ . For large α the function ψ is a scaled version of κ .

The results of the numerical investigation shown in Figure 1 seem to indicate that the Newton-type flow approximates movement by constant normal speed

$$\frac{\partial \Gamma}{\partial t} = c\mathbf{n},$$

or, in level-set formulation,

$$\phi_t + c|\nabla\phi| = 0.$$

The following calculation, however, shows that the constant c depends on the length of the level-set Γ . We consider a parametrized level-set of ϕ given by $\Gamma = \{(x_1(s), x_2(s)) : s \in [0, L]\}$ in two space dimensions. Integrating the left- and right-hand sides of (37) along Γ gives

$$\int_0^L \psi(x_1(s), x_2(s)) ds = \bar{\psi} L = \frac{2\pi}{\alpha},$$

i.e., the average speed

$$\bar{\psi} = \frac{1}{L} \int_0^L \psi(x_1(s), x_2(s)) ds = \frac{2\pi}{\alpha L} \quad (38)$$

of each level set depends on the overall length of the level set. Thus, we can expect to get a geometric evolution equation of the form

$$\frac{\partial \Gamma}{\partial t} = c(|\Gamma|)\mathbf{n},$$

with $c(|\Gamma|) \rightarrow \infty$ as $|\Gamma| \rightarrow 0$ in the limit $\alpha \rightarrow 0$. Note that short level sets move (and vanish) faster. On the other hand, large-scale structures (i.e., level-sets with large area) are persistent over longer time-intervals. It is obvious that, for level sets consisting of multiple connected components, the above discussion must be done separately for each connected component. The above discussion holds under the assumption that $\phi = b_\Gamma$. Thus, the flow (26) has the described qualitative behavior for small times if the initial level-set function ϕ_0 is a signed distance function. This property, however, is not maintained during the propagation of ϕ , and we see a different qualitative behavior if the evolution time is sufficiently large.

3.3 Is this Approach Trivial?

Let us consider an analogous approach where we replace (11) by the similar functional

$$K(\phi) = \int_{\mathbb{R}^n} |\nabla \phi|^2 d\mathbf{x}. \quad (39)$$

Obviously

$$K'(\phi) \psi = \frac{1}{2} \int_{\mathbb{R}^n} \langle \nabla \phi, \nabla \psi \rangle d\mathbf{x} \quad \text{and} \quad K''(\phi) (\psi, \psi) = \frac{1}{2} \int_{\mathbb{R}^n} \langle \nabla \psi, \nabla \psi \rangle d\mathbf{x}.$$

The optimality system for the determination of a Newton-type descent direction — analogous to (25) — then reads as

$$\frac{1}{2} \int_{\mathbb{R}^n} \langle \nabla \phi, \nabla \eta \rangle d\mathbf{x} + \frac{\lambda}{4} \int_{\mathbb{R}^n} \langle \nabla \psi, \nabla \eta \rangle d\mathbf{x} = 0$$

for all test functions η . This equation obviously has a solution $\psi = c\phi$ leading to the rather trivial flow equation

$$\phi_t = -\phi$$

which is of zeroth-order in $\mathbf{x}$. The solution $\phi(\mathbf{x}, t) = e^{-t}\phi_0(\mathbf{x})$ clearly diminishes the value of the functional (39).

The situation is not as simple for the non-quadratic functional $\tilde{J}$. Let us consider the situation $\phi = b_\Gamma$ and the formulation (35) with $\alpha = 0$. We get

$$-\Delta_\Gamma \psi = \Delta \phi,$$

or

$$-\operatorname{div}_\Gamma(\nabla_\Gamma \psi) = \operatorname{div}_\Gamma(\nabla \phi),$$

where $\operatorname{div}_\Gamma \mathbf{v} = \operatorname{div} \mathbf{v} - \mathbf{n}^t \cdot D\mathbf{v} \cdot \mathbf{n}$ is the tangential divergence of a vector field $\mathbf{v}$. This equation looks like we can again set $\psi = -\phi$ to obtain a solution. This is, however, not the case since $\nabla \phi$ is orthogonal to Γ , and therefore $\nabla_\Gamma \phi = 0$.

3.4 Relation to Inverse Scale Space Methods

Recently [4] the following system of equations was introduced

$$p_t = f - \phi \quad (40a)$$

$$p = -\operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \quad (40b)$$

for the smoothing of a noisy image f . The approach was called an inverse scale space method due to the property that — starting with zero — large features are restored first and fine structures and noise appear late in the evolution of the system (40). The approach can be seen as the continuous limit of an iterated Tikhonov regularization (or proximal point) algorithm for quadratic cost functionals, or as a continuous limit of Bregman iterations for the non-quadratic case as in (40). See also [23] for an analogous approach for linear problems. We shall show that (40) is closely related to the Newton-flow (26) although the starting point of the investigations in [4] is quite different from the approach presented here. To this aim, we write (40) in weak form

$$-\int_0^\infty \int_{\mathbb{R}^n} p \tilde{\eta}_t \, d\mathbf{x} \, dt = \int_0^\infty \int_{\mathbb{R}^n} (f - \phi) \tilde{\eta} \, d\mathbf{x} \, dt \quad (41)$$

for all test functions $\tilde{\eta} \in \mathcal{D}((0, \infty) \times \mathbb{R}^n)$, and

$$\int_{\mathbb{R}^n} p \eta \, d\mathbf{x} = \int_{\mathbb{R}^n} \frac{1}{|\nabla \phi|} \langle \nabla \phi, \nabla \eta \rangle \, d\mathbf{x} \quad (42)$$

for all $\eta \in \mathcal{D}(\mathbb{R}^n)$. Setting $\eta = \tilde{\eta}_t$ in (42) and integrating with respect to t gives

$$\int_0^\infty \int_{\mathbb{R}^n} p \tilde{\eta}_t \, d\mathbf{x} \, dt = \int_0^\infty \int_{\mathbb{R}^n} \frac{1}{|\nabla \phi|} \langle \nabla \phi, \nabla \tilde{\eta}_t \rangle \, d\mathbf{x} \, dt \quad (43)$$

for all $\tilde{\eta} \in \mathcal{D}((0, \infty) \times \mathbb{R}^n)$. Partial integration of the right-hand side of (43) with respect to t together with (41) yields

$$\begin{aligned} \int_0^\infty \int_{\mathbb{R}^n} \left[\frac{1}{|\nabla \phi|} \left(\langle \nabla \phi_t, \nabla \tilde{\eta} \rangle - \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \phi_t \right\rangle \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \tilde{\eta} \right\rangle \right) \right] \, d\mathbf{x} \, dt \\ = \int_0^\infty \int_{\mathbb{R}^n} (f - \phi) \tilde{\eta} \, d\mathbf{x} \, dt \end{aligned} \quad (44)$$

for all test functions $\tilde{\eta} \in \mathcal{D}((0, \infty) \times \mathbb{R}^n)$. Setting $\tilde{\eta}(\mathbf{x}, t) = \eta(\mathbf{x}) \zeta(t)$ and letting ζ vary throughout $\mathcal{D}((0, \infty))$, we conclude from (44) that

$$\int_{\mathbb{R}^n} \left[\frac{1}{|\nabla \phi|} \left(\langle \nabla \phi_t, \nabla \eta \rangle - \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \phi_t \right\rangle \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \eta \right\rangle \right) \right] d\mathbf{x} = \int_{\mathbb{R}^n} (f - \phi) \eta d\mathbf{x} \quad (45)$$

holds for all $\eta \in \mathcal{D}(\mathbb{R}^n)$. If we set $\psi = \phi_t$, we obtain the coupled system

$$\int_{\mathbb{R}^n} \left[\frac{1}{|\nabla \phi|} \left(\langle \nabla \psi, \nabla \eta \rangle - \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \psi \right\rangle \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \eta \right\rangle \right) \right] d\mathbf{x} = \int_{\mathbb{R}^n} (f - \phi) \eta d\mathbf{x} \quad (46a)$$

$$\phi_t = \psi \quad (46b)$$

for all test functions $\eta \in \mathcal{D}(\mathbb{R}^n)$. Thus, the system (46) is very similar to (26). The difference between the two is that the right hand side (the source term) in (26a) represents curvature whereas the right-hand side in (46a) represents the data fit $f - \phi$. Moreover, the zero-order regularization term on the left-hand side in (26a) is not present in (46a). The usual initial value for the flow (46) is $\phi_0 = 0$. Using this initial value, it is observed that ϕ evolves towards that given data function f with large features and texture appearing first and random noise appearing only in the late stages of the evolution.

We can also go in the reverse direction and rewrite the weak Newton flow (26) in a form similar to (40). We set again $p = -\operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right)$. Then p appears on the right-hand side of (30) and

$$p_t = -\operatorname{div} \left(\frac{1}{|\nabla \phi|} \left(\nabla \phi_t - \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \phi_t \right\rangle \frac{\nabla \phi}{|\nabla \phi|} \right) \right)$$

appears in the left-hand side of (30) if ψ is replaced by ϕ_t . Thus, we obtain

$$\begin{aligned} p_t + \alpha \frac{\phi_t}{|\nabla \phi|} &= -p \\ p &= -\operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \end{aligned}$$

for a formulation of the Newton flow which is analogous to (40).

3.5 Boundary Conditions

Any numerical solution of (33) requires to replace $\mathbb{R}^n$ by a bounded open domain D as domain of definition. In this case, boundary conditions for ψ have to be specified. We use the weak formulation

$$\begin{aligned} \int_D \frac{1}{|\nabla \phi|} \left(\langle \nabla \psi, \nabla \eta \rangle - \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \psi \right\rangle \left\langle \frac{\nabla \phi}{|\nabla \phi|}, \nabla \eta \right\rangle + \alpha \psi \eta \right) d\mathbf{x} \\ = \int_D \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \eta d\mathbf{x} \quad (47) \end{aligned}$$

for all test functions $\eta \in \mathcal{D}(\mathbb{R}^n)$.

Let $\boldsymbol{\nu}$ denote the outer unit normal to the set D . Application of Green's formula shows that the boundary condition is given in strong form as

$$\langle \nabla_\Gamma \psi, \boldsymbol{\nu} \rangle = 0 \quad (48)$$

on ∂D . Here ∇_Γ denotes the tangential derivative with respect to Γ . If $\boldsymbol{\nu}$ is not normal to Γ , equation (48) is in fact a homogenous Neumann-type boundary condition on $\Gamma \cap \partial D$ for the intrinsic equation (35).

Note that we have integrated the right hand side of (26a) by parts to obtain (47) but we have omitted the boundary term $-\int_{\partial D} \langle \mathbf{n}, \boldsymbol{\nu} \rangle \eta \, dS$. With this modification, (47) is not derived from the cost functional

$$\int_D |\nabla \phi| \, d\mathbf{x} \quad (49)$$

along the same (variational) reasoning as (26). If this was done, we end up with a non-homogeneous boundary condition

$$\langle \nabla_\Gamma \psi, \boldsymbol{\nu} \rangle = -\langle \mathbf{n}, \boldsymbol{\nu} \rangle |\nabla \phi|. \quad (50)$$

This type of boundary condition takes into account the effect of shrinking the area of level sets within D by pushing them out of D across ∂D . We choose the boundary condition (48) to avoid this type of influence of the boundary on the evolution of level-sets. Note also that the non-homogeneous boundary condition (50) introduces singularities at points $\mathbf{x} \in \Gamma \cap \partial D$ where $\boldsymbol{\nu} = \pm \mathbf{n}$. At these points the right-hand side of (50) is zero and the left-hand side is in general different from zero.

4 Numerical Examples

Our numerical examples exemplify the behavior of the system (26), i.e., we use the weak formulation of the Newton-type flow. For fixed ϕ , the degenerate elliptic equation (26a) is solved for ψ using bilinear tensor splines for the discretization of ψ . The singularity at $\nabla \phi = 0$ is treated replacing $|\nabla \phi|$ by $(|\nabla \phi|^2 + \epsilon^2)^{\frac{1}{2}}$ whenever $\nabla \phi$ occurs in the denominator. It turned out to be better to regularize not only $|\nabla \phi|$ in the denominator, but to replace $\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}$ by $\frac{\nabla \phi + \mathbf{e}^\epsilon}{|\nabla \phi + \mathbf{e}^\epsilon|}$ with a small vector $\mathbf{e}$ which pushes $\nabla \phi$ away from zero at all occurrences of $\mathbf{n}$ in (26a). Most of the examples are carried out with the geometric L^2 -regularization (21). Only at the end of this section we make a comparison between the formulations (21) and (22).

We begin the experimental study of properties of the flow (26) by comparing the propagations of a noise-free, non-convex level curve for different values of the regularization parameter α . By gradually increasing α , the characteristic of the flow changes from a global shrinking of level-sets with constant

speed to a local shrinking of level-sets by diminishing the curvature locally. We also mention that the admissible step-size of the flow decreases with increasing α , i.e., the more the flow resembles mean-curvature motion. The initial level-set function ϕ_0 is chosen as the signed distance function of the initial curve. The plots in Figure 2 show the zero level-sets of the level set function ϕ at every k^{th} iteration of the time-stepping algorithm (k between 20 and 80 for the individual choices of α), where the time-step is controlled by a fixed CFL-like criterion. The condensation of level-sets within certain time intervals is explained by the development of blocky structures in the level-set function ϕ . If a situation is reached, where blocky structures with sharp edges have occurred, the propagation is decelerated mainly due to the reduction of the admissible time-step which depends on $|\nabla\phi|$. In sections, where this influence of the time-step reduction is not yet active, an acceleration of the speed of propagation for smaller level-sets can be observed. This uneven spacing of level-lines for consecutive iteration numbers constitutes a structural difference between our flow (for small α) and flow with constant speed (i.e. construction of the signed distance function of an initial curve). Note that, for small α , the initially connected level set splits into two parts. This behavior is impossible for classical mean curvature flow. (See [21, 22].)

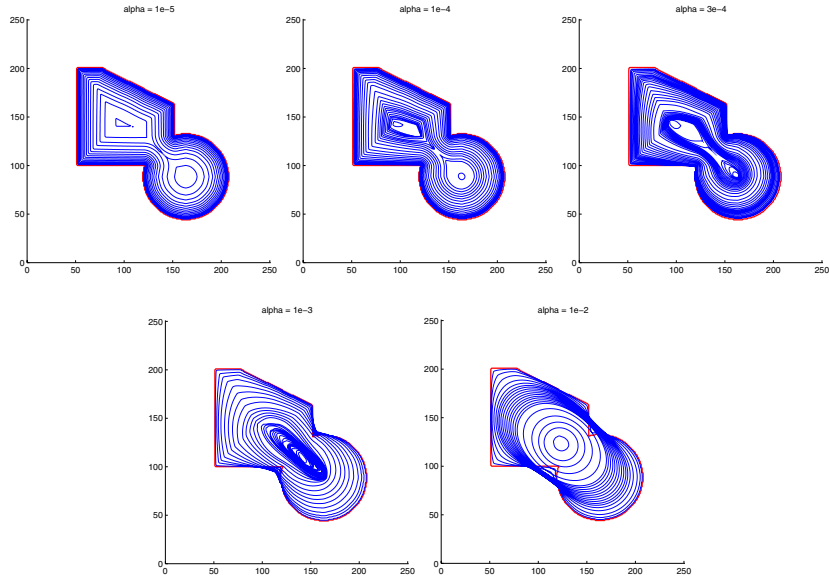


Fig. 2. Behavior of the Newton-type flow for $\alpha = 10^{-5}, 10^{-4}, 3 \cdot 10^{-3}, 10^{-3}, 10^{-2}$. Gradual interpolation between constant speed and mean curvature flow.

The next Figure 3 shows that the average speed of different level sets depends on their respective size with smaller level-sets moving faster. The

initial level-sets are drawn in red. In blue the zero-level set (consisting of multiple components of different size) is drawn every 15 time-steps. It can be clearly seen that the propagation accelerates the smaller the components get. We mention, that the situation is such that during the whole propagation the level-set function is still close to a signed-distance function and blocky structures have not yet developed.

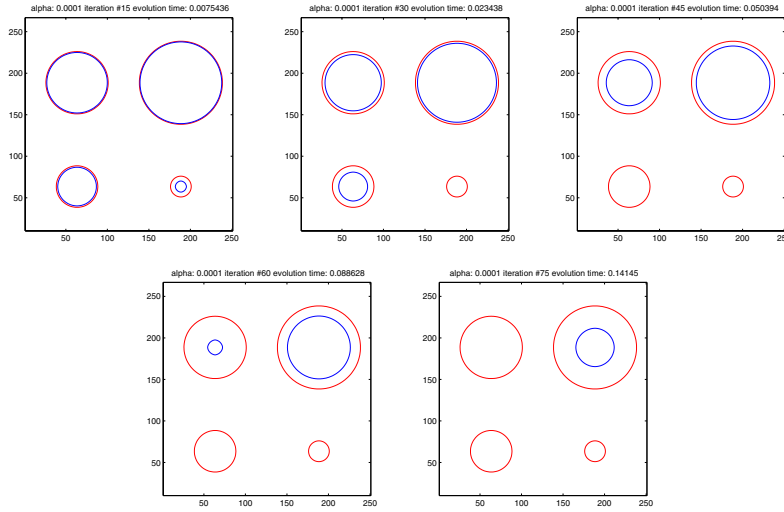


Fig. 3. Behavior of the Newton-type flow for small $\alpha = 10^{-4}$ and initial level-sets of different size. Smaller level-sets move faster.

In Figure 4 the behavior of the flow for oscillating initial data is illustrated. We have used the level-set function ϕ of the experiments shown in Figure 2 and overlayed it with a highly oscillating perturbation. The experiment is carried out with small $\alpha = 10^{-5}$. It is seen that the length of the curve is not reduced by local smoothing but by shrinking the oscillating structure globally.

Figure 5 shows the evolution of a real noise-free image under the Newton-type flow (26). It is seen that smooth structures are flattened out fast, whereas edges and texture are maintained very well. Note that the texture of the feathers and the straw-hat in the Lena image are still maintained (and even enhanced to some extent) when smoother structures as the nose or the structure in the left background have already vanished. Step edges as e.g., the frame of the mirror on the right-hand side of the image are maintained longer than smoother structures but not as long as textured structures as e.g., the straw-hat. Note also that the smaller bright structure in the lower left corner vanishes much earlier than the larger structure along the left edge of the image, although their initial brightness, contrast, and sharpness are very much

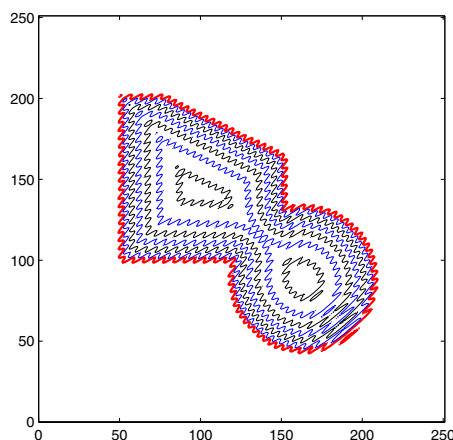


Fig. 4. Newton-type flow of a curve with highly oscillating curvature for $\alpha = 10^{-5}$. The length of the curve is reduced globally, not locally.

the same. Figure 6 is an enlarged version of two images in Figure 5. The persistent edges and textures are clearly visible. The flow reduces the overall contrast of the images from the initial range of gray values between 0 and 255 to the interval between 75 and 145 at iteration no. 1200. For better visibility the images have been scaled so that minimal intensity appears black and the maximal intensity appears white. Other numerical experiments with noisy images indicate that high frequency noise is removed from a level-set function (i.e. an image) more or less in the same amount as signal is removed. Thus, the Newton-flow — for small alpha — is not well suited for image denoising.

In the last Figure 7 a comparison between regularization by the geometric L^2 -norm (16) and the non-geometric L^2 -norm (18) is shown. The choice $\alpha = 10^{-3}$ has been made to find a situation where the regularization term has some influence on the behavior of the flow, but α is not large enough to change the characteristic of the flow strongly towards pure mean curvature or total variation flow. It is seen that the non-geometric version to the left has a certain preference for flat regions and blocky structures, whereas the geometric regularization to the right maintains smooth transitions of gray values slightly better.

5 Conclusion

We have introduced a nonlocal flow equation which reduces the area of level-sets of an initial function based on a gradient descent flow with respect to a variable metric constructed from the second derivative of the cost functional. We have investigated geometrical properties of the flow and we have compared our approach with the recently introduced inverse scale space method.

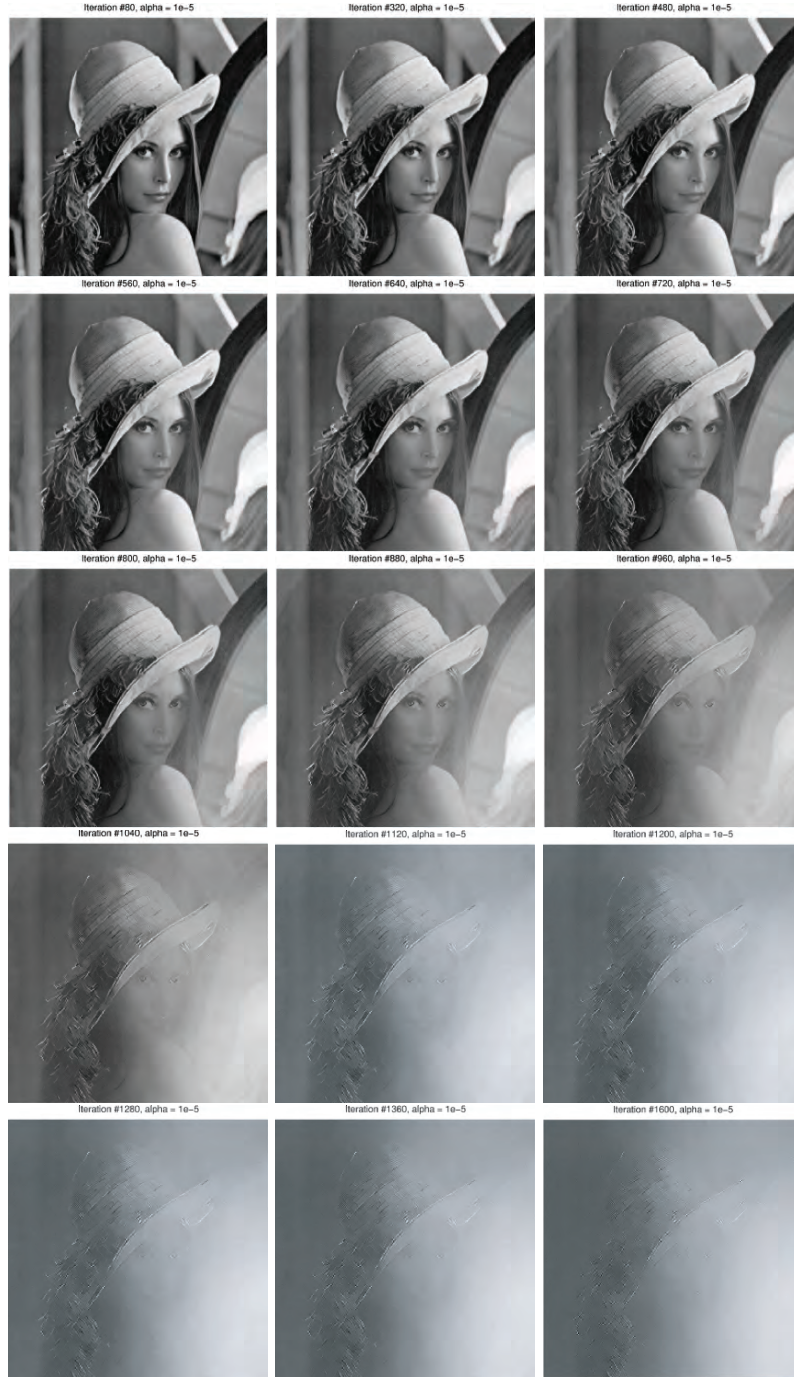


Fig. 5. Newton-type flow for the Lena image. $\alpha = 10^{-5}$. Smooth structures are removed, edges and texture are maintained

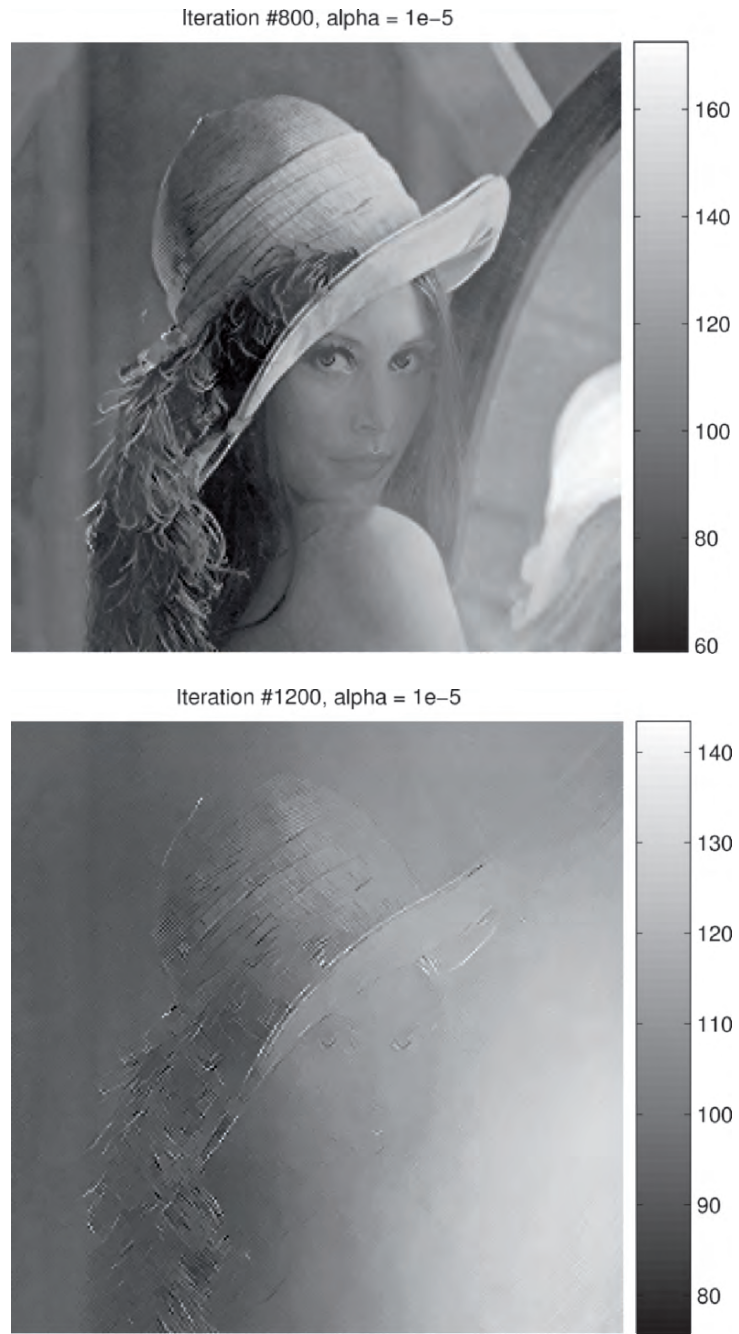


Fig. 6. Magnified Lena image at iterations 800 and 1200.

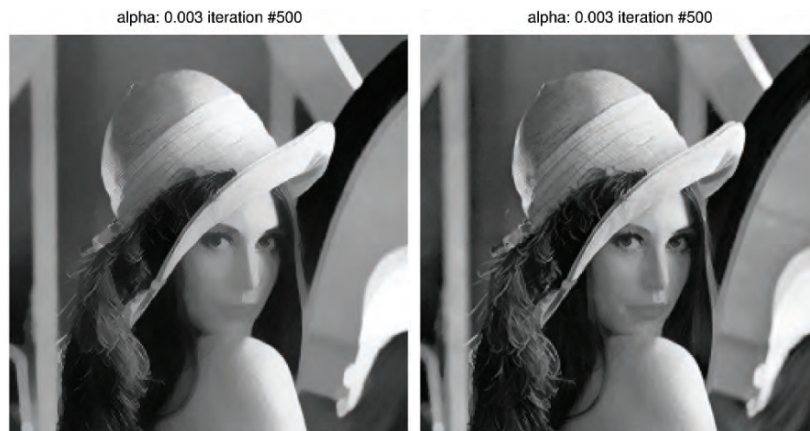


Fig. 7. Comparison between L^2 -regularization (left) and geometric regularization (right) for $\alpha = 10^{-3}$.

Numerical experiments were presented illustrating features of the flow for single propagating level-sets and for images for which all level-sets propagate simultaneously.

The practical applicability of the approach as it stands is probably limited to situations, where small structures are to be removed from an image but large structures should remain untouched. Our approach treats noise as large, wiggled level-sets which are spread over a certain area in space. Such level sets are shrunk very slowly, therefore noise is not removed from the image. The general approach of constructing a Newton-type flow for an arbitrary geometric functional in the level-set context has a variety of potential applications. Expected benefits are speed up of the minimization algorithm and the occurrence of interesting features in the flows due to the availability of non-local information.

References

1. F. Andreu, C. Ballester, V. Caselles, and J. M. Mazón. Minimizing total variation flow. *Differ. Integr. Equat.*, 14(3):321–360, 2001.
2. G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson. Image segmentation using active contours: calculus of variations or shape gradients? *SIAM J. Appl. Math.*, 63(6):2128–2154 (electronic), 2003.
3. V. Barbu. *Analysis and Control of Nonlinear Infinite Dimensional Systems*, volume 190 of *Mathematics in Science and Engineering*. Academic Press, Boston, 1993.
4. M. Burger, G. Gilboa, S. Osher, and J. Xu. Nonlinear inverse scale space methods. *Comm. Math. Sci.*, 4(1), 2006.
5. F. Cao. *Geometric curve evolution and image processing*, volume 1805 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2003.

6. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. J. of Comput. Vis.*, 22(1):61–79, 1997.
7. A. Chambolle. An algorithm for mean curvature motion. *Interfaces and Free Boundaries*, 6(2):195–218, 2004.
8. Y. G. Chen, Y. Giga, and S. Goto. Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations. *J. Differ. Geom.*, 33(3):749–786, 1991.
9. M. G. Crandall, H. Ishii, and P.-L. Lions. User’s guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Math. Soc. (N.S.)*, 27(1):1–67, 1992.
10. K. Deckelnick and G. Dziuk. Mean curvature flow and related topics. In *Frontiers in numerical analysis (Durham, 2002)*, Universitext, pages 63–108. Springer, Berlin, 2003.
11. K. Deckelnick and G. Dziuk. Numerical approximation of mean curvature flow of graphs and level sets. In *Mathematical aspects of evolving interfaces (Funchal, 2000)*, volume 1812 of *Lecture Notes in Math.*, pages 53–87. Springer, Berlin, 2003.
12. K. Deckelnick, G. Dziuk, and Ch. M. Elliott. Computation of geometric partial differential equations and mean curvature flow. *Acta Numer.*, 14:139–232, 2005.
13. M. C. Delfour and J.-P. Zolésio. *Shapes and geometries*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001. Analysis, differential calculus, and optimization.
14. D. C. Dobson and C. R. Vogel. Convergence of an iterative method for total variation denoising. *SIAM J. Numer. Anal.*, 34(5):1779–1791, 1997.
15. M. Droske and M. Rumpf. A level set formulation for Willmore flow. *Interfaces and Free Boundaries*, 6(3):361–378, 2004.
16. L. C. Evans and R. F. Gariepy. *Measure theory and fine properties of functions*. CRC Press, Boca Raton, FL, 1992.
17. L. C. Evans and J. Spruck. Motion of level sets by mean curvature. I. *J. Differ. Geom.*, 33(3):635–681, 1991.
18. L. C. Evans and J. Spruck. Motion of level sets by mean curvature. II. *Trans. Amer. Math. Soc.*, 330(1):321–332, 1992.
19. L. C. Evans and J. Spruck. Motion of level sets by mean curvature. III. *J. Geom. Anal.*, 2(2):121–150, 1992.
20. L. C. Evans and J. Spruck. Motion of level sets by mean curvature. IV. *J. Geom. Anal.*, 5(1):77–114, 1995.
21. M. Gage and R. S. Hamilton. The heat equation shrinking convex plane curves. *J. Differ. Geom.*, 23(1):69–96, 1986.
22. M. A. Grayson. The heat equation shrinks embedded plane curves to round points. *J. Differ. Geom.*, 26(2):285–314, 1987.
23. C.W. Groetsch and O. Scherzer. Non-stationary iterated Tikhonov-Morozov method and third-order differential equations for the evaluation of unbounded operators. *Math. Methods Appl. Sci.*, 23(15):1287–1300, 2000.
24. S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Conformal curvature flows: from phase transitions to active vision. *Arch. Rational Mech. Anal.*, 134(3):275–301, 1996.
25. W. Kühnel. *Differential geometry*, volume 16 of *Student Mathematical Library*. American Mathematical Society, Providence, RI, 2002. Curves—surfaces—manifolds, Translated from the 1999 German original by Bruce Hunt.

26. J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.
27. S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1): 12–49, 1988.
28. L.T. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithm. *Physica D*, 60:259–268, 1992.
29. J. Sokółowski and J-P. Zolésio. *Introduction to shape optimization*. Springer-Verlag, Berlin, 1992. Shape sensitivity analysis.
30. J. E. Solem and N. Chr. Overgaard. A geometric formulation of gradient descent for variational problems with moving surfaces. In Ron Kimmel, Nir Sochen, and Joachim Weickert, editors, *Scale space and PDE methods in computer vision. 5th international conference, Scale-Space 2005, Hofgeismar, Germany, April 7–9, 2005. Proceedings.*, Lecture Notes in Computer Science 3459, pages 419–430. Springer, 2005.
31. C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM J. Sci. Comput.*, 17(1):227–238, 1996. Special issue on iterative methods in numerical linear algebra (Breckenridge, CO, 1994).
32. A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum. A geometric snake model for segmentation of medical imagery. *IEEE Trans. Med. Imag.*, 16(3):199–209, 1997.

Chromaticity Denoising using Solution to the Skorokhod Problem

Dariusz Borkowski

Faculty of Mathematics and Computer Science, Nicolaus Copernicus University,
ul. Chopina 12/18, 87-100 Toruń, Poland. E-mail: dbor@mat.uni.torun.pl

Summary. Color image denoising methods based on the chromaticity-brightness decomposition are well-known for their excellent results. We propose a novel approach for chromaticity denoising using advanced techniques of stochastic calculus. We consider the Skorokhod problem associated with the solution of backward stochastic differential equation and an epsilon neighborhood of two dimensional sphere. BSDE is driven by a diffusion process corresponding to the image geometry. The proof of existence of a solution to such problems leads to a numerical scheme. We illustrate our method by synthetic and real examples.

Key words: chromaticity denoising, Skorokhod problem, backward stochastic differential equations

1 Introduction

The inverse problem of restoration of noisy image by automatic and reliable methods belongs to the most intensively studied topics of image processing. Various techniques of noise removal were proposed to tackle this problem. We may quote the linear filtering, variational/PDE-based approaches [1, 3, 4, 5, 6, 7, 9, 13, 14], wavelets theory and stochastic modeling (generally based on the Markov field theory). Most methods for the color images have been formulated on channel-by-channel and vectorial model. In this paper we study the restoration based on the chromaticity-brightness decomposition. This model is known to be closer to human perception of colors and gives good results. Many authors proposed to use norm constrained regularizing flows, in order to deal with chromaticity denoising [3, 4, 5, 6, 7, 14]. We are going to present a new stochastic method of chromaticity denoising which involves two advanced tools of stochastic analysis: backward stochastic differential equation [2, 8, 11, 12, 15, 18] and solution to the Skorokhod problem [16, 17].

First, we consider the problem of image denoising with values in $\mathbb{R}^n$. This problem is represented by backward stochastic differential equation (BSDE for short). The BSDE is driven by a diffusion process X corresponding to an image geometry. Since the image is defined on a bounded domain, the process X is considered as a process with reflection. The trajectory of the process Y which satisfies the BSDE, models the mechanics of the image denoising. The reconstructed pixel is the value of Y at time zero. This procedure is the starting point for a reconstruction of a chromaticity. In order to cope with this problem we solve the Skorokhod problem associated with the trajectory of the process Y and an epsilon neighborhood of the two dimension sphere ($\overline{S_\epsilon^2}$). The solution (which is also denoted by Y) is determined by a correction process K which has a bounded variation and increases only when the trajectory of Y is on the boundary of $\overline{S_\epsilon^2}$. Thus, it is possible to preserve good properties of the trajectory of the solution to BSDE. Similar to $\mathbb{R}^n$ case, the value of the new process Y at time zero is the reconstructed chromaticity.

2 Mathematical Preliminaries

2.1 Skorokhod Problem

Let $D \subset \mathbb{R}^n$ be a domain, with closure $\overline{D}$ and boundary ∂D . Suppose that a function $y_{(\cdot)}$ taking values in $\mathbb{R}^n$ is given and $y_0 \in \overline{D}$. It is often the case that one wishes to construct a function x with the following heuristic description. The starting point of y and x are the same, so $x_0 = y_0$. If y_t is in $\overline{D}$, then the evolution of x mimics y . If $x \in \partial D$ and imitating y would force x to exit $\overline{D}$ then the correction term is added. This term is the minimal push needed to keep x in $\overline{D}$. If imitating y does not result in the exit of x from $\overline{D}$, then correction term is unnecessary.

Situations where constraint mechanisms of this type occur are common in statistics, economics, queuing theory, telecommunication and computer networks. The Skorokhod problem provides a very useful definition which translates the heuristic description just given into precise mathematical term. The definition of the Skorokhod problem can be made general enough to cover a very wide variety of domains, constraints and input functions. When the mapping is well defined and sufficiently regular, it provides a convenient tool for the study of many types of constrained deterministic and stochastic problems.

Let $T > 0$ and $\mathbb{C}([0, T]; \mathbb{R}^n)$ denote the set of $\mathbb{R}^n$ valued continuous functions.

Definition 1. Let y be an element of the set $\mathbb{C}([0, T]; \mathbb{R}^n)$ such that $y_0 \in \overline{D}$. A pair $(x, k) \in \mathbb{C}([0, T]; \mathbb{R}^{2n})$ is said to be a solution to the Skorokhod problem associated with y and D if

- (i) $x_t = y_t + k_t$, $t \in [0, T]$,
- (ii) $x_t \in \overline{D}$, $t \in [0, T]$,

(iii) k is a function with bounded variation on $[0, T]$, $k_0 = 0$ and

$$k_t = \int_0^t n_s d|k|_s, |k|_t = \int_0^t 1_{\{x_s \in \partial D\}} d|k|_s, \quad t \in [0, T],$$

where $n_s = n(x_s)$ is a normal unit vector at $x_s \in \partial D$,

Existence and uniqueness of the solution to the Skorokhod problem was proved for sets which satisfies conditions (A) and (B) (see [16]), where

$$(A) \quad \exists_{r_0 \in (0, \infty]} \forall_{x \in \partial D} N_x = N_{x, r_0} \neq \emptyset,$$

$$(B) \quad \exists_{\delta > 0, \beta \geq 0} \forall_{x \in \partial D} \exists_{I_x, |I_x|=1} \forall v \in \bigcup_{y \in B(x, \delta) \cap \partial D} N_y < I_x, v \geq \frac{1}{\beta},$$

$$N_x = \bigcup_{r > 0} N_{x, r},$$

$$N_{x, r} = \{v \in \mathbb{R}^n : |v| = 1, B(x - rv, r) \cap D = \emptyset\},$$

$$B(x, r) = \{y \in \mathbb{R}^n : |x - y| < r\}, x \in \mathbb{R}^n, r > 0.$$

2.2 Elements of Stochastic Analysis

Definition 2. Let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space.

- (i) A stochastic process is a parametrized collection of random variables $X = \{X_t; t \in [0, T]\}$ defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ with values in $\mathbb{R}^n$. Note that for each fixed $\omega \in \Omega$ we can consider the function $t \rightarrow X_t(\omega)$ which is called a trajectory of X and is denoted by $X(\omega)$.
- (ii) A filtration is a nondecreasing family $(\mathcal{F}_t) = \{\mathcal{F}_t; t \in [0, T]\}$ of sub- σ -fields of $\mathcal{F}$ i.e., $\mathcal{F}_s \subseteq \mathcal{F}_t \subseteq \mathcal{F}$ for $0 \leq s < t \leq T$. A filtration $(\mathcal{F}_t^X)$ is generated by process X if, $\mathcal{F}_t^X = \sigma(X_s; 0 \leq s \leq t)$.
- (iii) The stochastic process X is adapted to the filtration $(\mathcal{F}_t)$ (X is $(\mathcal{F}_t)$ adapted) if, for each $t \in [0, T]$, X_t is $\mathcal{F}_t$ - measurable random variable.
- (iv) The stochastic process X is progressively measurable with respect to the filtration $(\mathcal{F}_t)$ (X is $(\mathcal{F}_t)$ - progressively measurable) if, for each $t \in [0, T]$ and $\mathcal{A} \in \mathcal{B}(\mathbb{R}^n)$ ($\mathcal{B}(U)$ denote the smallest σ -field containing all open sets of a topological space U), the set $\{(s, \omega); 0 \leq s \leq t, \omega \in \Omega, X_s(\omega) \in \mathcal{A}\}$ belongs to the product σ -field $\mathcal{B}([0, t]) \otimes \mathcal{F}_t$ ($\mathcal{F} \otimes \mathcal{G} = \sigma(A \times B; A \in \mathcal{F}, B \in \mathcal{G})$).

Definition 3. Let Y be an $(\mathcal{F}_t)$ adapted process with continuous trajectories, $Y_0 \in \overline{D}$. We say that a pair (X, K) of $(\mathcal{F}_t)$ adapted processes solves the Skorokhod problem associated with Y and D if for almost every $\omega \in \Omega$, $(X(\omega), K(\omega))$ is a solution to the Skorokhod problem associated with $Y(\omega)$ and D .

Let W be a Wiener process, $x_0 \in \overline{D}$ and $\sigma : [0, T] \times \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^m$.

Definition 4. Let Y be an $(\mathcal{F}_t)$ adapted process. A pair $(X, K^{\overline{D}})$ of $(\mathcal{F}_t)$ adapted processes is called a solution of reflected SDE

$$X_t = x_0 + \int_0^t \sigma(t, X_t) dW_t + K_t^{\overline{D}}, \quad t \in [0, T], \quad (1)$$

if $(X, K^{\overline{D}})$ is a solution to the Skorokhod problem associated with

$$Y_t = x_0 + \int_0^t \sigma(s, X_s) dW_s, \quad t \in [0, T],$$

and D .

The proof of existence and uniqueness of solution to reflected SDEs for convex sets can be found in [17].

Let $(\mathcal{F}_t^W)$ be filtration generated by W , $\xi \in L^2(\Omega, \mathcal{F}_T, P, \mathbb{R}^k)$.

Definition 5. The solution of the equation BSDE associated with ξ is a pair of $(\mathcal{F}_t^W)$ - progressively measurable processes (Y_t, Z_t) with values in $\mathbb{R}^k \times \mathbb{R}^{k \times l}$ satisfying the following conditions:

$$E \left[\int_0^T |Z_s|^2 ds \right] < \infty,$$

$$Y_t = \xi - \int_t^T Z_s dW_s, \quad t \in [0, T]. \quad (2)$$

See [15] for the proof of existence and uniqueness of solution to BSDEs.

3 Stochastic Representation of Solution to the Heat Equation

Let us begin with formulation of the problem. Let D be a bounded, convex domain in $\mathbb{R}^2$, $u : \overline{D} \rightarrow \mathbb{R}^n(S^2)$ be an original image and $u_0 : \overline{D} \rightarrow \mathbb{R}^n(S^2)$ be the observed image of the form

$$u_0 = u + \eta,$$

where η stands for a white Gaussian noise. We assume that u and u_0 are appropriately regular. We are given u_0 , the problem is to reconstruct u .

3.1 The $\mathbb{R}^n$ Case

Before presenting a general model, we will illustrate our ideas by giving a simple example. We will construct a model which is equivalent to a commonly used filter, namely, the convolution of the noise image with the two-dimensional Gaussian mask. The construction of our model is an appropriate construction of the processes X and Y . We suppose for a moment that the image is a function defined on the whole plane and define

$$\begin{cases} X_t = W_t^x, & t \in [0, T], \\ Y_t = u_0(X_T) - \int_t^T Z_s dW_s, & t \in [0, T], \end{cases} \quad (3)$$

where W_t^x is a Wiener process starting from $x \in \overline{D}$. Equation (3) we can simplify to the form

$$\begin{cases} X_t = W_t^x, & t \in [0, T], \\ Y_0 = u_0(X_T) - \int_0^T Z_s dW_s = \\ \quad = \mathbb{E}u_0(X_T) = \int_{\mathbb{R}^2} G_{\sqrt{T}}(x - y)u_0(y) dy, \end{cases} \quad (4)$$

where $G_\sigma(x) = \frac{1}{2\pi\sigma^2}e^{-\frac{|x|^2}{2\sigma^2}}$ is the two-dimensional Gaussian mask.

A value of the process Y at time $t = 0$ is the reconstructed pixel $u(x)$. We got the image which is the convolution of the noise image with two-dimensional Gaussian mask.

While discussing the above example, we assumed that the image is the function given on the whole plane. Since we want to consider the image as a function defined on the bounded, convex set, we have to introduce a new assumption for the process X . We assume that the process X is a stochastic process with reflection with values in $\overline{D}$. In this case process X is a Wiener process with reflection, which we can write as

$$\begin{cases} X_t = W_t^x + K_t^{\overline{D}}, & t \in [0, T], \\ Y_t = u_0(X_T) - \int_t^T Z_s dW_s, & t \in [0, T]. \end{cases} \quad (5)$$

3.2 The S^2 Case

Now we will show, how to transform the model of reconstruction of the image with values in $\mathbb{R}^n$ into a model of reconstruction of chromaticity.

Let u_0 be the image with values in $\mathbb{R}^3$ (RGB representation)

$$u_0(x) = (R(x), G(x), B(x)) \in \mathbb{R}^3. \quad (6)$$

Each color vector $u_0(x)$ can be split into its norm and its unit direction vector, i.e.,

$$u_0^I(x) = \sqrt{R(x)^2 + G(x)^2 + B(x)^2} \in \mathbb{R}, \quad (7)$$

$$u_0^C(x) = \left(\frac{R(x)}{u_0^I(x)}, \frac{G(x)}{u_0^I(x)}, \frac{B(x)}{u_0^I(x)} \right) \in S^2, \quad (8)$$

where u_0^I is an intensity part and u_0^C is a chromaticity part. It is well known that acting separately on these two different color characteristics allows to reconstruct the noisy image more precisely than RGB representation (see [6]). We will explore the problem of the chromaticity reconstruction.

Let $x \in \overline{D}$ be a fixed point of the image. Given u_0^C , we have to reconstruct a value $u^C(x) \in S^2$.

To solve this problem with using the model (5) we have to introduce a boundary for values of the process Y . We would like that process Y has values in two-dimensional sphere. In order to achieve it we exploit a solution to the Skorokhod problem. Since the Skorokhod problem was considered in sets with non empty interior we demand that the process Y has values in the set

$$\overline{S_\epsilon^2} = \{y \in \mathbb{R}^3; 1 - \epsilon \leq |y| \leq 1 + \epsilon\}. \quad (9)$$

This condition is important for us because guarantees that the trajectories of the correction process K are functions with bounded variation. In this way we consider the following model of chromaticity denoising.

$$\begin{cases} X_t = W_t^x + K_t^{\overline{D}}, & t \in [0, T], \\ Y_t = u_0^C(X_T) - \int_t^T Z_s dW_s + K_t^{\overline{S_\epsilon^2}} - K_t^{\overline{S_\epsilon^2}}, & t \in [0, T]. \end{cases} \quad (10)$$

In the next section we will explain precisely the meaning of this equation. It should be observed that the above equation is not reflected backward stochastic differential equation (RBSDE). Existence and uniqueness of solution to RBSDEs was proved in [11] but only for convex sets.

3.3 Existence of a Solution

Relation between the processes $Y, Z, K^{\overline{S_\epsilon^2}}$ is described in the following theorem:

Theorem 1. *Let D be the set satisfying conditions (A) and (B). Let $(\mathcal{F}_t^W)$ be a filtration generated by W , $\xi \in L^2(\Omega, \mathcal{F}_T, P, \mathbb{R}^k)$, $\xi(\omega) \in \overline{D}$ for almost every $\omega \in \Omega$. Then there exists a triple of processes (Y, Z, K) with values in $\mathbb{R}^k \times \mathbb{R}^{k \times l} \times \mathbb{R}^k$, which satisfies conditions:*

- (i) Z is $(\mathcal{F}_t^W)$ - progressively measurable,
- (ii)

$$E \left[\int_0^T |Z_s|^2 ds \right] < \infty,$$

- (iii) Y has continuous trajectories, $Y_t \in \overline{D}$,
- (iv) trajectories of the process K are continuous functions with bounded variation,
- (v)

$$Y_t = \xi - \int_t^T Z_s dW_s + K_T - K_t, \quad t \in [0, T]. \quad (11)$$

Proof. We prove the theorem in two steps. First we define processes which satisfy (i) - (iv), next we show that the formula (11) holds.

Let Z be a process which satisfies the following BSDE

$$\tilde{Y}_t = \xi - \int_t^T Z_s dW_s, \quad t \in [0, T].$$

Such process exists, is $(\mathcal{F}_t^W)$ - progressively measurable and satisfies condition (ii) (see [15]).

Putting $\tilde{Y}_t = \tilde{Y}_{T-t}$ and using the fact that trajectories of the process $\tilde{Y}$ are continuous we have $\tilde{Y}(\omega) \in \mathbb{C}([0, T]; \mathbb{R}^k)$, $\tilde{Y}_0(\omega) = \xi(\omega) \in \overline{D}$ for almost every $\omega \in \Omega$.

For the process $\hat{Y}$ and filtration $(\mathcal{F}_t^W) = \{\mathcal{F}_t; \mathcal{F}_t = \mathcal{F}_T^W, t \in [0, T]\}$ we solve the Skorokhod problem (see [16]) and we find processes $(\check{Y}, \check{K})$ such that

$$\begin{aligned} \check{Y}_t &= \hat{Y}_t - \check{K}_t, & t \in [0, T], \\ \check{Y}_t &\in \overline{D}, & t \in [0, T]. \end{aligned}$$

Moreover the process $\check{K}$ has trajectories with bounded variation, $\check{K}_0 = 0$ and the process $\check{Y}$ has values in $\overline{D}$.

Let us define

$$\begin{aligned} Y_t &= \check{Y}_{T-t}, & t \in [0, T], \\ K_t &= \check{K}_{T-t}, & t \in [0, T]. \end{aligned}$$

Note that the equation (11) holds. Indeed,

$$Y_t = \check{Y}_{T-t} = \hat{Y}_{T-t} - \check{K}_{T-t} = \tilde{Y}_t - K_t = \xi - \int_t^T Z_s dW_s + K_T - K_t.$$

□

Remark 1. The set $S_\epsilon^2 = \{y \in \mathbb{R}^3; 1 - \epsilon < |y| < 1 + \epsilon\}$, satisfies conditions (A) and (B).

4 Image Denoising

The model we constructed in the previous section is equivalent to the convolution of the noise image with the two-dimensional Gaussian mask. This filter removes noise and blurs edges. In this section we provide a construction which has the following properties:

- (i) noise is removed,
- (ii) image has sharp edges.

4.1 Local Geometry of the Image – Gray Level Images

First, we shall construct a model for gray levels images.

In a neighborhood of an edge, the image exhibits a strong gradient. In order to preserve this edge, we should diffuse along it ([13]). We assume that the process X has the form

$$X_t = x + \int_0^t \sigma_1(s, X_s) dW_s + K_t^{\overline{D}}, \quad (12)$$

where

$$\sigma_1(t, X_t) = \lambda_1(|\nabla u_0(X_t)|) \begin{bmatrix} -\frac{u_{0x_2}(X_t)}{|\nabla u_0(X_t)|}, & 0 \\ \frac{u_{0x_1}(X_t)}{|\nabla u_0(X_t)|}, & 0 \end{bmatrix}, \quad \lambda_1(s) > 0. \quad (13)$$

At locations where the variations of the brightness are weak (low gradient), we would like to encourage smoothing, the same in all direction. We expect that process X will have the property of the Wiener process. This condition may be achieved by imposing

$$X_t = x + \int_0^t \sigma_2(s, X_s) dW_s + K_t^{\overline{D}}, \quad (14)$$

where the diffusion coefficient has the form

$$\sigma_2(t, X_t) = \lambda_2(|\nabla u_0(X_t)|) \begin{bmatrix} -\frac{u_{0x_2}(X_t)}{|\nabla u_0(X_t)|}, & \frac{u_{0x_1}(X_t)}{|\nabla u_0(X_t)|} \\ \frac{u_{0x_1}(X_t)}{|\nabla u_0(X_t)|}, & \frac{u_{0x_2}(X_t)}{|\nabla u_0(X_t)|} \end{bmatrix}, \quad \lambda_2(s) > 0. \quad (15)$$

Combining the above assumptions we can write X as

$$X_t = x + \int_0^t \sigma(s, X_s) dW_s + K_t^{\overline{D}}, \quad (16)$$

where

$$\sigma(t, X_t) = \begin{bmatrix} -\lambda_1(|\nabla u_0(X_t)|) \frac{u_{0x_2}(X_t)}{|\nabla u_0(X_t)|}, & \lambda_2(|\nabla u_0(X_t)|) \frac{u_{0x_1}(X_t)}{|\nabla u_0(X_t)|} \\ \lambda_1(|\nabla u_0(X_t)|) \frac{u_{0x_1}(X_t)}{|\nabla u_0(X_t)|}, & \lambda_2(|\nabla u_0(X_t)|) \frac{u_{0x_2}(X_t)}{|\nabla u_0(X_t)|} \end{bmatrix},$$

$$\lim_{s \rightarrow 0} \lambda_1(s) > 0,$$

$$\lim_{s \rightarrow 0} \lambda_2(s) > 0,$$

$$\lim_{s \rightarrow 0} \frac{\lambda_1(s)}{\lambda_2(s)} = 1,$$

$$\lim_{s \rightarrow \infty} \lambda_1(s) > 0,$$

$$\lim_{s \rightarrow \infty} \lambda_2(s) = 0.$$

As an example we can use functions which is shown in Figure 1. In this

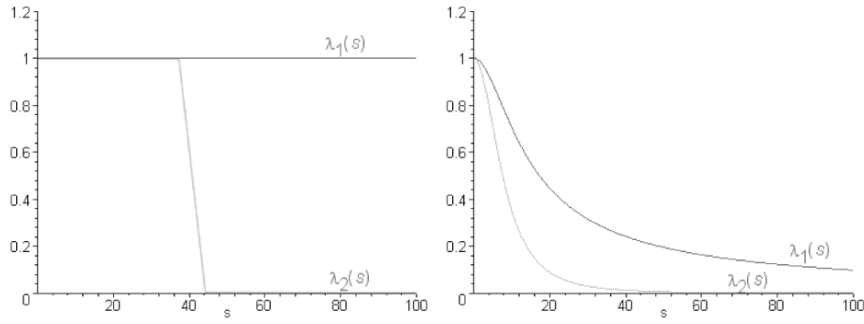


Fig. 1. Examples of functions λ_1 and λ_2 .

situation we have the following model of denoising for gray levels images:

$$\begin{cases} X_t = x + \int_0^t \sigma(s, X_s) dW_s + K_t^{\overline{D}}, & t \in [0, T], \\ Y_t = u_0(X_T) - \int_t^T Z_s dW_s, & t \in [0, T]. \end{cases} \quad (17)$$

4.2 Local Geometry of the Image – RGB Images

Denoising RGB images by direct applications of gray levels method to each component separately does not bring satisfying results ([9]). For RGB images as for gray levels images the diffusion process X must be driven in common way for all colors. Multivalued geometry for images was proposed by Di Zenzo in [10]. He considers a multivalued image u as a $2D \rightarrow nD$ vector field and looks for the local variations of the vector norm $\|du\|^2$,

$$\|du\|^2 = du^T du = \|u_{x_1}\|^2 dx_1^2 + 2u_{x_1}^T u_{x_2} dx_1 dx_2 + \|u_{x_2}\|^2 dx_2^2, \quad (18)$$

i.e.,

$$\|du\|^2 = dx^T G dx, \quad (19)$$

where

$$G = \begin{pmatrix} \sum_{i=1}^n (u_{x_1}^i)^2 & \sum_{i=1}^n u_{x_1}^i u_{x_2}^i \\ \sum_{i=1}^n u_{x_1}^i u_{x_2}^i & \sum_{i=1}^n (u_{x_2}^i)^2 \end{pmatrix}. \quad (20)$$

Positive eigenvalues λ_+ , λ_- are the maximum and the minimum of $\|du\|^2$ respectively, while the orthogonal eigenvalues θ_+ and θ_- are the corresponding variation orientations ([9]). We use this geometry for construction of a model for color images.

Let u_0 have the form

$$u_0(x_1, x_2) = (R(x_1, x_2), G(x_1, x_2), B(x_1, x_2)). \quad (21)$$

Then we have the following model of denoising

$$\begin{cases} X_t = x + \int_0^t \sigma(s, X_s) dW_s + K_t^{\overline{D}}, & t \in [0, T], \\ Y_t = u_0(X_T) - \int_t^T Z_s dW_s, & t \in [0, T], \end{cases} \quad (22)$$

where

$$\begin{aligned} \sigma(t, X_t) &= \begin{bmatrix} -\lambda_1(N(X_t))\theta_+^2(X_t), & \lambda_2(N(X_t))\theta_+^1(X_t) \\ \lambda_1(N(X_t))\theta_+^1(X_t), & \lambda_2(N(X_t))\theta_+^2(X_t) \end{bmatrix}, \\ \Delta &= (R_{x_1}^2 + G_{x_1}^2 + B_{x_1}^2 - R_{x_2}^2 - G_{x_2}^2 - B_{x_2}^2)^2 + \\ &\quad + 4(R_{x_1}R_{x_2} + G_{x_1}G_{x_2} + B_{x_1}B_{x_2})^2, \\ \lambda_{\pm} &= \frac{R_{x_1}^2 + G_{x_1}^2 + B_{x_1}^2 + R_{x_2}^2 + G_{x_2}^2 + B_{x_2}^2 \pm \sqrt{\Delta}}{2}, \\ \nu_{\pm} &= \begin{bmatrix} 2(R_{x_1}R_{x_2} + G_{x_1}G_{x_2} + B_{x_1}B_{x_2}) \\ R_{x_2}^2 + G_{x_2}^2 + B_{x_2}^2 - R_{x_1}^2 - G_{x_1}^2 - B_{x_1}^2 \pm \sqrt{\Delta} \end{bmatrix}, \\ \theta_{\pm} &= \frac{\nu_{\pm}}{|\nu_{\pm}|}, \\ N &= \sqrt{\lambda_+ - \lambda_-}. \end{aligned}$$

4.3 Chromaticity Denoising

If we are given a model for images with values in $\mathbb{R}^n$, we can easily construct the model for images with values in $\overline{S_\epsilon}$. We need to introduce a boundary for

values of the process Y in analogy with stochastic representation of solution to the heat equation. In this case we consider

$$\begin{cases} X_t = x + \int_0^t \sigma(s, X_s) dW_s + K_t^{\overline{D}}, & t \in [0, T], \\ Y_t = u_0^C(X_T) - \int_t^T Z_s dW_s + K_T^{\overline{S_\epsilon^2}} - K_t^{\overline{S_\epsilon^2}}, & t \in [0, T], \end{cases} \quad (23)$$

where

$$\sigma(t, X_t) = \begin{bmatrix} -\lambda_1(N(X_t))\theta_+^2(X_t), & \lambda_2(N(X_t))\theta_+^1(X_t) \\ \lambda_1(N(X_t))\theta_+^1(X_t), & \lambda_2(N(X_t))\theta_+^2(X_t) \end{bmatrix},$$

θ_+ , N are determined by geometry of the chromaticity u_0^C . Functions λ_1 and λ_2 are the same as in the previous section.

5 A Numerical Scheme

Numerical schemes for BSDE are described in [2, 8, 12, 18] for example. Discrete approximation of solutions to the Skorokhod problem and reflected SDE can be found in [16]. Using these numerical schemes we propose the following method for reconstruction of chromaticity:

$$\begin{aligned} \tilde{Y}_{t_i}^n &= E[u_0^C(X_T^x) | \mathcal{F}_{t_i}], & 0 = t_0 < t_1 < \dots < t_{n-1} = T, \\ Y_{t_{n-1}}^n &= \tilde{Y}_{t_{n-1}}^n, \\ Y_{t_i}^n &= \Pi_{\overline{S_\epsilon^2}}(Y_{t_{i+1}}^n + \tilde{Y}_{t_i}^n - \tilde{Y}_{t_{i+1}}^n), & i = 0, 1, \dots, n-2, \end{aligned} \quad (24)$$

where $\Pi_{\overline{S_\epsilon^2}}$ denotes projection on $\overline{S_\epsilon^2}$.

In Figure 2 we present denoising of a color image. The color image is decomposed into RGB channels, from which we can extract brightness and the chromaticity. We add Gaussian noise to the chromaticity vectors only, leaving the brightness unchanged. Chromaticity has been denoised by running S-BSDE filter solving (23) with $T = 10$, $\epsilon = 0.02$.

References

1. G. Aubert and P. Kornprobst. *Mathematical problems in image processing*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2002. Partial differential equations and the calculus of variations, With a foreword by Olivier Faugeras.



Fig. 2. Chromaticity denoising. Top-bottom: original, noisy, denoised. (Color images in Figure A.12.)

2. V. Bally. Approximation scheme for solutions of BSDE. In *Backward stochastic differential equations (Paris, 1995–1996)*, volume 364 of *Pitman Res. Notes Math. Ser.*, pages 177–191. Longman, Harlow, 1997.
3. V. Caselles, G. Sapiro, and B. Tang. Diffusion on general data on non-flat manifolds via harmonic maps theory: the direction diffusion case. *Int. J. Comput. Vis.*, 36(2):149–161, 2000.
4. V. Caselles, G. Sapiro, and B. Tang. Color image enhancement via chromaticity diffusion. *IEEE Trans. Image Process.*, 10(5):701–707, 2001.
5. T. Cecil, S. Osher, and L. Vese. Numerical methods for minimization problems constrained to S^1 and S^2 . *J. Comput. Phys.*, 198(2):567–579, 2004.
6. T. Chan, S. H. Kang, and J. Shen. Total variation denoising and enhancement of color image based on the CB and HSV color models. *J. Vis. Comm. Image Represent.*, 12(4):422–435, 2001.
7. T. Chan and J. Shen. Variational restoration of nonflat image features: models and algorithms. *SIAM J. Appl. Math.*, 61(4):1338–1361 (electronic), 2000/01.
8. D. Chevance. Numerical methods for backward stochastic differential equations. In *Numerical methods in finance*, Publ. Newton Inst., pages 232–244. Cambridge Univ. Press, Cambridge, 1997.
9. R. Deriche and D. Tschumperlé. Diffusion PDE’s on vector-valued images: local approach and geometric viewpoint. *IEEE Signal Process. Mag.*, 19(5):16–25, 2002.
10. S. Di Zenzo. A note on the gradient of a multi-image. *Comput. Vis. Graph. Image Process.*, 33(1):116–125, 1986.
11. A. Gégout-Petit and E. Pardoux. Équations différentielles stochastiques rétrogrades réfléchies dans un convexe. *Stochast. Stochast. Rep.*, 57(1-2):111–128, 1996.
12. J. Ma, P. Protter, J. San Martín, and S. Torres. Numerical method for backward stochastic differential equations. *Ann. Appl. Probab.*, 12(1):302–316, 2002.
13. J. Malik and P. Perona. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, 1990.
14. S. J. Osher and L. A. Vese. Numerical methods for p -harmonic flows and applications to image processing. *SIAM J. Numer. Anal.*, 40(6):2085–2104 (electronic) (2003), 2002.
15. E. Pardoux. Backward stochastic differential equations and viscosity solutions of systems of semilinear parabolic and elliptic PDEs of second order. In *Stochastic analysis and related topics, VI (Geilo, 1996)*, volume 42 of *Progr. Probab.*, pages 79–127. Birkhäuser Boston, Boston, MA, 1998.
16. Y. Saisho. Stochastic differential equations for multidimensional domain with reflecting boundary. *Probab. Theor. Relat. Field.*, 74(3):455–477, 1987.
17. H. Tanaka. Stochastic differential equations with reflecting boundary condition in convex regions. *Hiroshima Math. J.*, 9(1):163–177, 1979.
18. J. Zhang. A numerical scheme for BSDEs. *Ann. Appl. Probab.*, 14(1):459–488, 2004.

Improved 3D Reconstruction of Interphase Chromosomes Based on Nonlinear Diffusion Filtering

Jan Hubený, Pavel Matula, Petr Matula¹, and Michal Kozubek

Masaryk University, Faculty of Informatics, Centre for Biomedical Image Analysis, Botanická 68a, 602 00 Brno, The Czech Republic. E-mail: {xhubeny, pam, pem, kozubek}@fi.muni.cz, url: <http://lom.fi.muni.cz/>

Summary. A recently published method for 3D reconstruction of interphase chromosomes based on the fast marching method was extended. A kind of nonlinear diffusion filtering, namely *balanced forward-backward* diffusion filtering, was added to the preprocessing phase of the algorithm. The method for finding an optimal arrival time was also modified. The nonlinear diffusion filtering can suppress inhomogeneities in chromosome staining while edges in images are preserved. This leads to better performance for chromosomes that are close to each other.

Key words: Nonlinear diffusion filtering, PDE based image processing, interphase chromosomes, biomedical application

1 Introduction

Genetic information is coded by double stranded DNA, which is deposited in cell nucleus in the form of chromosomes. Spatial organization of DNA and proteins in the cell nucleus and its functional relations are poorly known. Such knowledge is necessary for the full understanding of mechanisms and events in cell nuclei, which in turn can serve for the detection of abnormal states related to serious diseases. One of the reasons why this knowledge is still very limited is the absence of sophisticated and reliable image processing methods that could be used for automatic image analysis tasks. Analysis of a large number of objects is particularly important to ensure statistical significance of results. Image processing algorithms must be able to run automatically with minimal or no interaction with the operator.

Recent evidence has demonstrated that chromosomes occupy distinct domains in the cell nucleus, called chromosome territories [3, 4]. Each territory can be considered as a connected, variably-shaped, three-dimensional structure which is mutually exclusive from other territories. The territories are

often visualized by means of fluorescence in situ hybridization and the stained specimen is imaged using a confocal fluorescence microscope.

Confocal microscopy images of cells are often extremely noisy. Image denoising and enhancement is therefore an indispensable step towards achieving high-quality results. Conventional low-pass filters are usually sufficient for the suppression of statistical background noise (e.g. photon-shot noise, readout noise). However, other types of noise are present in the image, for instance, target object labeling can lead to inhomogeneities in the object intensities or small amount of fluorescent dyes can remain in improper places due to incomplete washing. All these phenomena complicate unsupervised image segmentation.

The following approaches were applied for chromosome territory reconstruction in the past. A computational geometry method based on Voronoi tessellation [1] was adapted for 3D reconstruction of interphase chromosomes in [5, 6]. A method based on local thresholding and mathematical morphology has also been used for chromosome segmentation [8]. Recently a method based on the well-known fast marching method has been proposed [12]. We propose an extension of the latter method based on nonlinear diffusion that improves its reliability. The extension is described below. Evaluation on real image data is also presented.

1.1 Input Data

The algorithm was studied on the following material. Targets in biological material (HL-60 blood cells) were visualized by fluorescence in situ hybridization. The chromatin of cells (occupies the whole volume of the nuclei) was stained by DAPI (blue colour). The chromosomes 9 were stained by Rhodamine (red colour) and chromosomes 22 by FITC (green colour).

The images of visualized targets were acquired using fully automated high-resolution cytometry system in the Laboratory of optical microscopy, Masaryk university Brno [8] (Zeiss Axiovert 100S inverted fluorescence microscope equipped with a CARV confocal module based on a Nipkow spinning disc). Specimen was observed through a PlanApochromat 63 $\times$ /1.4 oil immersion objective.

A stack of 40 2D images (parallel optical sections) was captured with a high-quality digital CCD camera for ten chosen stage positions (fields of view) for each colour. The CCD chip of the camera had 1300 $\times$ 1030 pixels; pixel size was 6.7 μ m. The dynamic range of the camera was 12 bits but only 8-bit integer was used for pixel intensity storage. The axial (z) step between two optical sections was 0.2 μ m. The lateral (x, y) step is given by the magnification power of the objective and the pixel size of the CCD chip and it was $\approx$ 0.1 μ m.

Each field of view typically contained tens of cells. For each 3D image also a maximal intensity projection image in the axial direction over all slices, called auto-focus (AF) image, was computed.

Large 3D input images covering the whole microscope field of view were segmented into small sub-images containing only one cell nucleus per image. An algorithm for cell nucleus segmentation based on a local thresholding [8] was applied on the AF chromatin images and a proper bounding box for each cell nucleus was computed. For details see [12].

2 Improved Reconstruction of Interphase Chromosomes

2.1 Algorithm

The proposed algorithm is an improvement of previously published method, which used the fast marching algorithm for the 3D reconstruction of interphase chromosomes [12]. The so-called *balanced forward-backward (BFB)* diffusion filtering [7] was added to the preprocessing phase of the algorithm. The method for finding the optimal arrival time was also modified. The input images have been processed in two steps:

Image Enhancement

The goal of this step was to suppress the noise while preserving edges in each of the sub-images, which were produced by the cell segmentation algorithm (see Section 1.1). Inhomogeneous signals of small amount of fluorescent dyes, which remain in improper places due to incomplete washing, should be suppressed as well. We used nonlinear diffusion filtering [13, 2, 14, 7] to fulfill this task.

The nonlinear diffusion filter has the following structure in m dimensions. Let $\Omega \in \mathbb{R}^m$ denote the m -dimensional image domain and $f : \Omega \rightarrow \mathbb{R}$ an initial grey-scale image. The nonlinear diffusion filter calculates a filtered image ($u(\mathbf{x}, t)$) of $f(\mathbf{x})$ as a solution of diffusion equation

$$\partial_t u = \operatorname{div}(g(|\nabla u| \nabla u)) \quad (1)$$

considering $f(\mathbf{x})$ as initial condition

$$u(\mathbf{x}, 0) = f(\mathbf{x}) \quad (2)$$

with reflecting boundary conditions

$$\partial_n u = 0 \quad \text{on} \quad \delta\Omega \quad (3)$$

where n denotes the outer normal to the image boundary $\delta\Omega$. The diffusion time t serves as a scale parameter, larger values of t give more simplified ("cartoon-like") images $u(\mathbf{x}, t)$. The diffusivity function g should be decreasing and nonnegative in order to smooth relatively homogenous regions while preserving significant edges.

From the wide family of nonlinear diffusion filters we used the BFB [7] filtering because of its properties: it removes oscillations, it preserves and

even enhances edges, and there is no additional contrast parameter for the diffusivity g function to tune.

The diffusivity function g in BFB filtering is set to

$$g(|\nabla u|) = \frac{1}{|\nabla u|^p} \quad p = 2 \quad (4)$$

To avoid numerical problems, which arise when the gradient magnitude $|\nabla u|$ gets close to zero, we used bounded version of the diffusivity function g .

$$g(|\nabla u|) = \frac{1}{(|\nabla u|^2 + \epsilon^2)} \quad (5)$$

We applied the semi-implicit AOS (additive operator splitting) scheme for the numerical implementation. The AOS scheme was proposed in [10, 11] and later independently rediscovered in [17]. This scheme is efficient even for small values of ϵ^2 (between 0.001 and 0.01). We could set approximately 20 times larger time step than in common explicit scheme and still meet our accuracy requirements.

The sub-images were diffused by BFB filter with bounded diffusivity function g with $\epsilon^2 = 0.001$ to time $t = 30$. The diffusion was numerically approximated by 10 iterations of AOS scheme with time step equal to 3.0. The diffusion time t was set by hand in order to get reasonably enhanced images (without noise and unwashed dyes, with equalized intensities within objects). The time step was set as large as possible in order to get the results quickly. Naturally, the AOS scheme is unconditionally stable. However, we have observed that the typical directional errors of this scheme become apparent when using larger time steps. The pixel spacing was assumed to be equal to one in all dimensions. The size of sub-images was approximately $96 \times 96 \times 40$ voxels. The results of the filtering are illustrated in Fig. 1.

Chromosome Territory Reconstruction

The well-known fast marching algorithm [15, 16] was applied for the chromosome territory reconstruction in each diffused sub-image. Eight corners of the sub-image were taken as a starting contour. The idea was to initialize the contour outside the objects. As the contour is marching through the data it slows down in points with a high gradient magnitude and waits there (regardless of the topology changes) for the contour passing the points with gentle gradient magnitude. We have used standard equation

$$F(\mathbf{x}) = \frac{1}{1 + |\nabla \mathcal{G}_\sigma * I(\mathbf{x})|} \quad (6)$$

for the speed function computation. The speed function was computed from diffused sub-images convolved with $3 \times 3 \times 3$ Gaussian kernel with variance $\sigma = 1.5$. The spatial step $h_i = 0.01$ was assumed in all directions. These

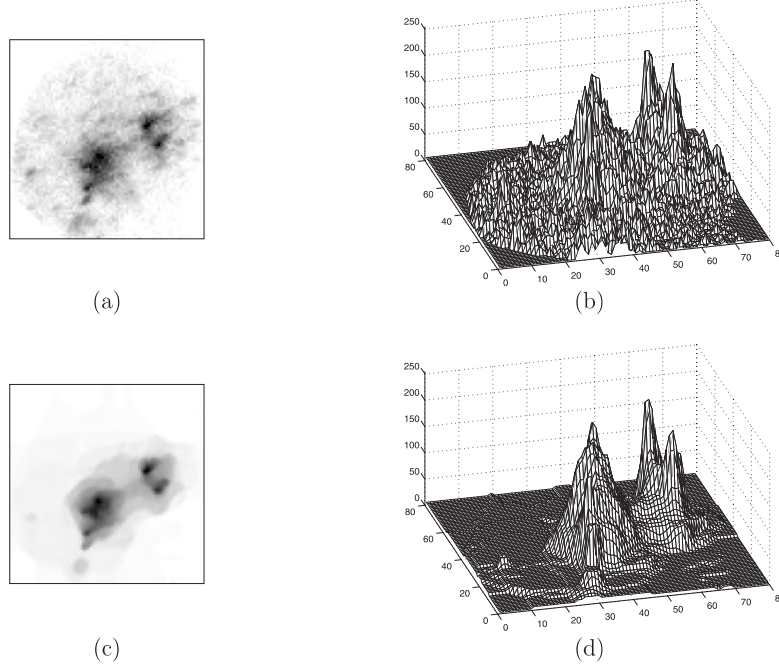


Fig. 1. Three-dimensional chromosome images are filtered with three-dimensional BFB filter in the first step of algorithm. (a) One slice from 40 slices of a typical input sub-image. (c) The same slice after diffusion with BFB filter. The intensity values in (a) and (c) are inverted for visualization purposes. (b),(d) Isometric representation of the slice before and after diffusion filtering. Note, that the diffusion filter reduces the noise, while preserving significant edges.

parameters were set by hand in order to slightly eliminate potential staircasing effect of the BFB filtering (the σ parameter) and to get high values in the image gradient (spatial step h) and therefore to get rapidly decreasing speed function (6). The final surface was appropriately stretched in the end for visualization purposes. The resulting surface of chromosome territories was visualized using the marching cube algorithm [9]. For a typical result see Fig. 2.

We slightly modified the method for finding the optimal arrival time because the original method sometimes produce over-segmented results. An optimal contour level was computed from a histogram of arrival time function $T(\mathbf{x})$. The histogram was constructed from integer part (floor) of function $T(\mathbf{x})$ and it mapped integer level t to the number of grid points which were visited between time t and time $t + 1$. The number of such grid points is related to the size (surface area) of the object defined by the level t . The goal was to find a level where the contour movement was very slow, i.e. the change



Fig. 2. An example of a final 3d computer model of reconstructed chromosome territories. The input sub-image of chromosome 22 pair in HL-60 nucleus was diffused with BFB filter (10 iterations with time step 3.0). The diffusion was computed with the AOS scheme. The fast marching algorithm was applied on diffused sub-image for the 3D reconstruction. Arrival time for which the proper level best approximated the objects boundary was computed automatically using histogram analysis. Finally, the proper level was visualized using the marching cube algorithm.

of contour size was small (i.e. the first derivative of histogram was minimal). The arrival time of the desired boundary was always between times 20 and 30, because the studied objects had similar sizes and the speed function was almost zero only near real edges due to the diffusion process. The level (between levels 20 and 30) whose second derivative was minimal was taken as the optimal level in our experiments. The histogram was smoothed by Gaussian kernel of size 7 and $\sigma = 0.5$ at the beginning.

2.2 Evaluation and Discussion

The proposed algorithm, which was described in Section 2, was successfully applied to our data (Section 1.1). The improved algorithm was tested on the same data as the previously published method [12] in order to easily compare the results. The testing sample comprises 25 confocal images of chromosome territories that were randomly selected from large image set (approximately hundred of cells), see [12] for details. We used only the green channel (chromosome 22) for our tests and comparisons. We have run the original as well as the improved version of the algorithm and compared the results.

The results for each sub-image were examined by an expert. The expert had to decide how many chromosome territories were in the nucleus according to the input image (usually two) and whether the algorithm found accurate boundary of the territories. The expert studied input data, superposition of the final model onto the input data and 3D computer model of chromosome territories.

We have realized, that the results of both versions of algorithm can be divided into three categories:

- The algorithm found the chromosome territories correctly.
- The algorithm located the territories accurately, but it found some small additional objects too.
- The algorithm found only one joined object, instead of two mutually exclusive territories.

We discuss the results of both algorithms in more details according to these three categories now. See also Table 1.

Table 1. Comparison of the original and the improved algorithm. The tests were performed on 25 individual sub-images. The values represent the percentage of the following three cases. First row: Sub-images processed without any problems. Second row: Reconstruction with small additional objects. Third row: Reconstruction with joined chromosome territories. Note that the additional objects and the joined territories could occur simultaneously in one image. Therefore, the sum in each column can exceed 100%.

	Original algorithm	Improved algorithm
No problem	20%	44%
Additional objects	64%	44%
Joined territories	28%	16%

Eleven sub-images were analyzed without any problem with the improved algorithm. Both the number of found territories and the boundary position agreed with expert's opinion. In comparison, there were only 5 sub-images analyzed without any problems using the original algorithm. An example of a typical final 3D reconstruction of chromosome territories of a cell nucleus is shown in Fig. 2. The projection of this final 3D model onto the input data (not diffused) is shown in Fig. 3 and 4.

The improved algorithm found more mutually exclusive territories than the expert in 11 cases. The magnitude of the additional objects was always smaller than the magnitude of the correctly located territories. Nevertheless the additional objects could be easily removed according to their size in a post-processing phase. The position of the contour agreed with expert opinion in all cases. The original algorithm found additional objects in 16 sub-images.

The improved algorithm found one joined object instead of two mutually exclusive territories in 4 cases. It occurred when the two territories were too close to each other. The original algorithm found one joined object in 7 input sub-images (in several cases both algorithms found also one or two small additional objects, which should and could be removed). The improved algorithm separated the territories better in cases, where the expert could see the weak boundary between close objects. Note that the expert often was not sure about the boundary between the close objects.

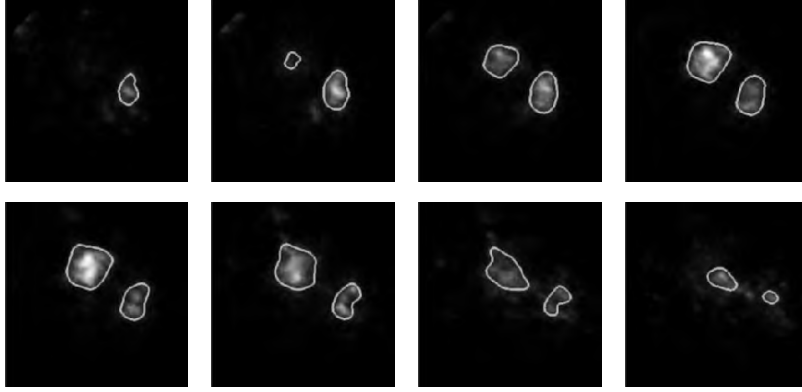


Fig. 3. Projection of the final model from Fig. 2 onto the input (not diffused) data is shown for 8 xy slices (top) $z = 18, 20, 22, 24$; (bottom) $z = 36, 38, 30, 32$.

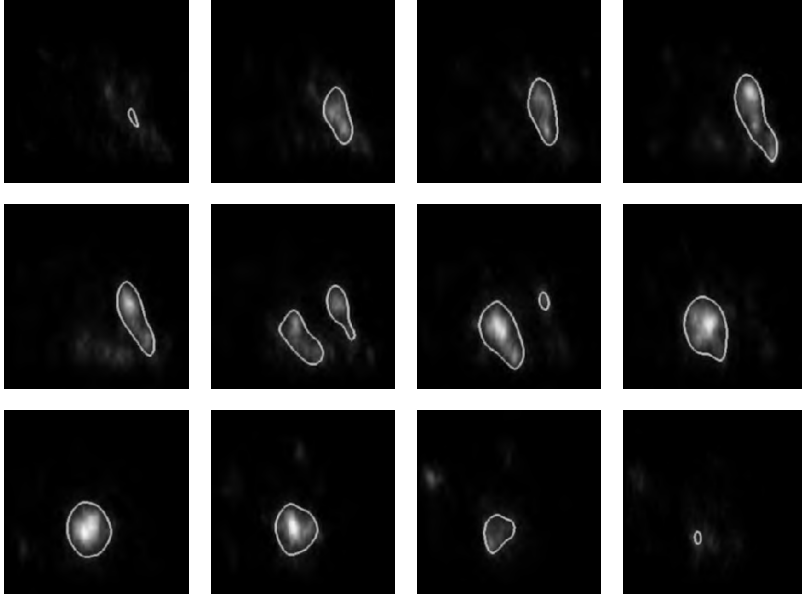


Fig. 4. Projection of the final model from Fig. 2 onto the input (not diffused) data is shown for 8 xz slices (top) $y = 30, 33, 36, 39$; (middle) $y = 42, 45, 48, 51$; (bottom) $y = 54, 57, 60, 62$

The sub-images of one cell nucleus had the average size of $96 \times 96 \times 40$ voxels. We used common PC workstation (Intel Pentium 4 2.6 GHz, Linux 2.6.5) in our experiments. 10 iterations of the BFB filter took 1.8 seconds in average, subsequent processing by the fast marching method took 1.14 seconds in average on the same machine.

3 Conclusion

Nonlinear diffusion filters can significantly improve the reliability of the 3D reconstruction of chromosome territories. More than 40% of problematic cases (chromosomes that were close to each other) were separated due to the enhanced preprocessing step. The method was unsuccessful only in cases where the expert was also not able to separate two neighbouring territories.

Although chromosome territories were automatically segmented using the fast-marching method in the filtered images, other methods could also be applied directly on the filtered images with success (e.g. a thresholding method).

Acknowledgment

This work was supported by the Ministry of Education of the Czech Republic (Grants No. MSM0021622419 and No. LC535) and by the Grant Agency of the Czech Republic (Project No. 204/03/D034).

References

1. E. Bertin, F. Parazza, and J. M. Chassery. Segmentation and measurement based on 3D Voronoi diagram: application to confocal microscopy. *Comput. Med. Imag. Graph.*, 17(3):175–182, 1993.
2. F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.*, 29(1):182–193, 1992.
3. T. Cremer and C. Cremer. Chromosome territories, nuclear architecture and gene regulation in mammalian cells. *Nat. Rev. Genet.*, 2(4):292–301, April 2001.
4. T. Cremer, A. Kurz, R. Zirbel, S. Dietzel, B. Rinke, E. Schröck, M. R. Speicher, U. Mathieu, A. Jauch, P. Emmerich, H. Scherthan, T. Reid, C. Cremer, and P. Lichter. Role of chromosome territories in the functional compartmentalization of the cell nucleus. *Cold Spring Harbor Symp Quantitative Biology*, 58:777–792, 1993.
5. R. Eils, E. Bertin, K. Saracoglu, B. Rinke, E. Schröck, F. Parazza, Y. Usson, M. Robert-Nicoud, E. H. K. Stelzer, J. M. Chassery, T. Cremer, and C. Cremer. Application of confocal laser microscopy and three-dimensional Voronoi diagrams for volume and surface estimates of interphase chromosomes. *J. Microsc.*, 177(2):150–161, February 1995.

6. R. Eils, S. Dietzel, E. Bertin, E. Schröck, M. R. Speicher, T. Ried, M. Robert-Nicoud, T. Cremer, and C. Cremer. Three-dimensional reconstruction of painted human interphase chromosomes: active and inactive X chromosome territories have similar volumes but differ in shape and surface structure. *J. Cell Biol.*, 135(6):1427–1440, December 1996.
7. S. L. Keeling and R. Stollberger. Nonlinear anisotropic diffusion filtering for multiscale edge enhancement. *Inverse Probl.*, 18:175–190, 2002.
8. M. Kozubek, S. Kozubek, E. Bártová E. Lukášová, M. Skalníková, Pavel Matula, Petr Matula, P. Jirsová, A. Cafourková, and I. Koutná. Combined confocal and wide-field high-resolution cytometry of FISH-stained cells. *Cytometry*, 45:1–12, 2001.
9. W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Computer Graphics (SIGGRAPH '87)*, volume 21, pages 163–169, 1987.
10. T. Lu, P. Neittaanmaki, and X. C. Tai. A parallel splitting up method and its application to navier-stokes equations. *Appl. Math. Lett.*, 4(2):25–29, 1991.
11. T. Lu, P. Neittaanmaki, and X. C. Tai. A parallel splitting up method for partial differential equations and its application to navier-stokes equations. *RAIRO Math. Model. and Numer. Anal.*, 26(6):673–708, 1992.
12. P. Matula, J. Hubený, and M. Kozubek. Fast marching 3d reconstruction of interphase chromosomes. In *CVAMIA*, LNCS 3117, pages 385–394. Springer-Verlag, 2004.
13. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, 1990.
14. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. In *Proceedings of the eleventh annual international conference of the Center for Nonlinear Studies on Experimental mathematics : computational issues in nonlinear science*, pages 259–268, Amsterdam, The Netherlands, The Netherlands, 1992. Elsevier North-Holland, Inc.
15. J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proc. Nat'l Academy of Sciences*, 93:1591–1595, 1996.
16. J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, 2nd edition, 1999.
17. J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. Image Process.*, 7(3):398–410, 1998.

Part III

Image Segmentation

Some Recent Developments in Variational Image Segmentation

Tony Chan¹, Mark Moelich², and Berta Sandberg³

¹ UCLA Mathematics Department chan@math.ucla.edu *

² Aerospace Corporation mark.c.moelich@aero.org

³ TechFinity, Inc. bsand@math.ucla.edu **

Summary. This survey paper discusses some recent developments in variational image segmentation and active contours models. Our focus will be on region-based models implemented via level-set techniques, typified by the Chan–Vese (CV) model [11]. The CV algorithm can be interpreted as a level-set implementation of the piecewise constant Mumford–Shah segmentation model and has been quite widely used.

We will first present the basic CV algorithm and an extension to piecewise smooth approximations. We also discuss a recent development in convexifying the CV model to guarantee convergence to a global minimizer. Next, we discuss extensions to handle multi-channel images, including a vector-valued CV model [9], texture segmentation [10], object tracking in video [41], image registration [40], and a logic segmentation framework [49]. Then we discuss multiphase extensions to handle segmentation into an arbitrary number of regions, including the method of Vese and Chan [61] and recent developments of memory efficiency algorithms such as the piecewise constant level set method (PCLSM) of Tai et al. [36] and the multi-layer method of Chung and Vese [13].

Finally, we discuss numerically efficient methods that attempt to compute the optimal segmentation much faster than the original gradient-descent PDE-based method. These methods include the direct pointwise optimization method of Song and Chan [55], an operator-splitting method by Gibou and Fedkiw [26], and a threshold dynamics method by Esedoglu and Tsai [19].

Key words: image segmentation, active contours without edges, Mumford–Shah, level sets, multi-phase, multi-channel, tracking, registration

*This work was supported by Office of Naval Research contract N00014-06-1-0345, National Science Foundation contract DMS-0610079, and National Institute of Health as part of the Center for Computational Biology at UCLA contract U54 RR021813

**This work was supported by Missile Defense Agency Small Business Technology Transfer Program contract HQ0006-05-C-7263

1 Introduction

Image segmentation is the process of partitioning an image into regions. Each region has a consistent trait throughout that is different from other regions in the image. Some common traits that have been captured are intensity, color, and texture. Once a decision is made on the desired traits, various segmentation methods are available to reach this goal.

This paper will focus on variational image segmentation and active-contour models and algorithms, which share the common feature that they define optimal segmentation as a minimizer of an objective function that generally depends on the given image and the traits that are used to identify the different segmented regions. The Euler–Lagrange equation of these models can often be described using a partial differential equation, which is iterated until it reaches steady state.

A contour is introduced into the image and is evolved until steady state thereby dividing the image into regions, see Figure 1. A very powerful and popular method for representing the contour is the level-set method originally developed by Osher and Sethian [45], which represents the contour implicitly as a particular (usually the zero) level of a (level-set) function. The main advantage of this representation is that topological changes, such as merging and pinching off of contours can be captured naturally through smooth changes to the level-set function.

In this paper, we will focus primarily on region-based (rather than edge-based) segmentation models. A prototypical example, and the primary one we will discuss in this paper, is the Chan–Vese “Active Contour Without Edges” model [11], which seeks the desired segmentation as the best piecewise constant approximation to a given image. The Chan–Vese model can be interpreted as a level-set implementation of the piecewise-constant special case of the more general Mumford–Shah segmentation model [43].

Due to its simplicity and robustness, the Chan–Vese model has become quite popular and has been adopted in many applications. As a result, a number of generalizations have been developed to improve both its applicability and efficiency. A natural generalization is to segmentation of multi-channel images. Initially, a vector valued method was used with an application in texture segmentation [10]. This was followed by an important conceptual generaliza-

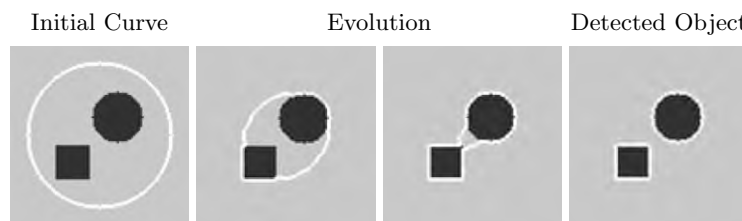


Fig. 1. Evolution of a contour around objects.

tion to a logic framework allowing the user to use any logical combination of information in each channel to obtain the desired segmentation. Further extensions include object tracking in video sequences in the presence of clutter, registration of images to identify key objects, and color segmentation that can identify an object in an image with an arbitrary combination of colors.

Another direction of generalization to the basic Chan–Vese model is to multiphase models, which allow the segmentation of the image into arbitrary (> 2) regions. A natural, but inefficient, generalization is to use one level-set function for each phase, taking care to avoid overlap and uncovered regions. Various attempts have been made to improve on this basic approach. The multiphase method of Vese and Chan [61] only needs $\log_2 n$ level-set functions to represent n regions, without any need to avoid overlap and uncovered regions, drastically improving the efficiency. More recently, Tai et al. [36] and Chung and Vese [13] have developed novel level-set methods that use only one level-set function to represent an arbitrary number of regions. We will review these methods in this paper.

A final direction of generalization is to improve the computational efficiency of these variation segmentation models. The typical approach of gradient flow (i.e., marching the Euler–Lagrange PDE to steady state) usually takes a long time to converge. A variety of methods have been developed to speed this up. One approach is to treat the models as a discrete optimization problem whose solution is the association of each pixel to a particular region. Song and Chan [55] proposed a direct optimization algorithm, which has the surprising property that for noiseless two-phase images the optimal solution can be provably obtained with only one sweep over the pixels. Gibou and Fedkiw [26] use an operator-splitting approach of treating the data term and the regularization (or curvature) term of the Euler–Lagrange equation in two separate steps, each of which can be computed very efficiently. Finally, Esetoglu and Tsai [19] use a threshold dynamics approach to obtain an efficient implementation. These methods will be discussed further in the paper.

The outline of the paper is as follows. Active-contour methods, and in particular the Chan–Vese model, are introduced in Section 2. In Section 3, we discuss multi-channel generalizations and in Section 4 we discuss multiphase generalizations. In Section 5, we discuss efficient implementations. Conclusions and possible directions for future research are given in Section 6.

2 Active Contours Methods

There are various schemes to deform the contour to the edges of an object. A quick summary and references for active contours using edge-detection algorithms are give below. For the rest of the paper we address active contours without edges, as written by Chan and Vese [11].

2.1 Classic Active Contours

A classic approach to active contour models is to use the gradient of the image u_0 to locate the edges of the object. Typically, an edge function is used that is positive inside homogeneous regions and strictly zero on the edges. Using this edge detection function, a functional is minimized with respect to contour C ,

$$\inf E(C) = \int_C |C'(s)|^2 ds + \lambda \int_C g(|\nabla u_0(C(s))|)^2 ds, \quad (1)$$

where g is an edge-detection function. This model is by Caselles, Kimmel, and Sapiro and similar work by Kass, Witkin, and Terzopolous [8, 30]. The model cannot handle automatic topology changes of the contour, and depends on the parameterization of the curve.

In problems of curve evolution, including snakes and active contours, the level-set method of Osher and Sethian [45] has been used extensively, because it allows for automatic topology changes, cusps and corners; moreover, the computations are made on a fixed rectangular grid. Using this approach, geometric active-contour models, using a stopping edge-function, have been proposed in [7], and also in [38]. These models are based on the theory of curve evolution and geometric flows. The evolving curve moves by mean curvature, but with an extra factor in the speed, the stopping edge-function. Therefore, the curve stops on the edges, where the edge-function vanishes. An example of edge-functions used is given by:

$$g(|\nabla u_0|) = \frac{1}{1 + |\nabla(G_\sigma * u_0)|^2},$$

where g is a positive and decreasing function, such that $\lim_{t \rightarrow \infty} g(t) = 0$. The image u_0 is first convolved with the Gaussian $G_\sigma(x, y) = \sigma^{-1/2} \exp^{-|x^2+y^2|/4\sigma}$, especially for the cases where u_0 is noisy. In practice, g is never zero on the edges, and therefore the evolving curve may not stop on the desired boundary. To overcome this problem, a new model has been proposed in [8], as a problem of geodesic computation in a Riemann space, according to the metric g . This gives an added term that increases the attraction of the evolving curve towards the boundary of the object, and is of special help when the boundary has high variations on its gradient values. For another related approach, see also [31].

These models use the gradient of a smoother version of the image u_0 , to detect edges. If the image is noisy, the smoothing in the edge-function has to be strong, thus blurring edge features, or a pre-processing has to be implemented, to remove the noise.

2.2 Active Contours without Edges

The Chan–Vese active-contour model without edges proposed in [11] does not use the stopping edge-function g to find the boundary. The stopping term is

based on Mumford–Shah [43] segmentation techniques. The equation for the Mumford–Shah in (u, C) is obtained by minimizing the functional:

$$E(u, C) = \int_{\Omega} (u - u_0)^2 dx + \mu \text{length}(C).$$

While the functional itself is elegant, in practice it is difficult to find a solution as the functional is non-convex, and has an unknown C . Various solutions have been proposed. One solution uses region growing, minimizing the Mumford–Shah functional using greedy algorithms [39, 32]. Elliptical approximations embed the contour C in a 2D phase-field function [1]. The Mumford–Shah functional has also been calculated using a statistical framework [66].

Let Ω be a bounded open subset of $\mathbb{R}^2$, with $\partial\Omega$ the boundary. Let u_0 be a given image such that $u_0 : \overline{\Omega} \rightarrow \mathbb{R}$. Let $C(s) : [0, 1] \rightarrow \mathbb{R}^2$ be a piecewise parameterized C^1 curve.

We choose a method with the following form:

$$\inf_{c^+, c^-, C} F(c^+, c^-, C),$$

where

$$\begin{aligned} F(c^+, c^-, C) = & \mu |C| + \lambda^+ \int_{in(C)} |u_0 - c^+|^2 dx \\ & + \lambda^- \int_{out(C)} |u_0 - c^-|^2 dx, \end{aligned} \quad (2)$$

where $|C|$ denotes the length of C , c^+ and c^- are constant unknowns representing the “average” value of u_0 inside and outside the curve, respectively. The parameters $\mu > 0$, and $\lambda^+, \lambda^- > 0$, are weights for the regularizing term and the fitting term, respectively.

Minimizing the fitting error in (2), the model approximates the image u_0 with a piecewise-constant function, taking only two values, namely c^+ and c^- , and with one edge C , the boundary between these two constant regions. The object to be detected will be given by one of the regions, and the curve C will be the boundary of the object. The additional length term is a regularizing term, and has a scaling role. If μ is large, only larger objects are detected, while for small μ , objects of smaller size are also detected. Because the model does not make use of a stopping edge-function based on the gradient, it can detect edges both with and without a gradient as can be seen in Figure 2. It is well known that (2) can be viewed as a special case of the Mumford–Shah segmentation [43].

We rewrite the original model (2) in the level-set formulation. Let the evolving curve C be embedded as the zero level set of a Lipschitz continuous function ϕ , i.e., $C(\phi) = \{(x, y) \in \Omega : \phi(x, y) = 0\}$, with ϕ having opposite signs on each side of C . Following [65] and [11], the energy can be written as:

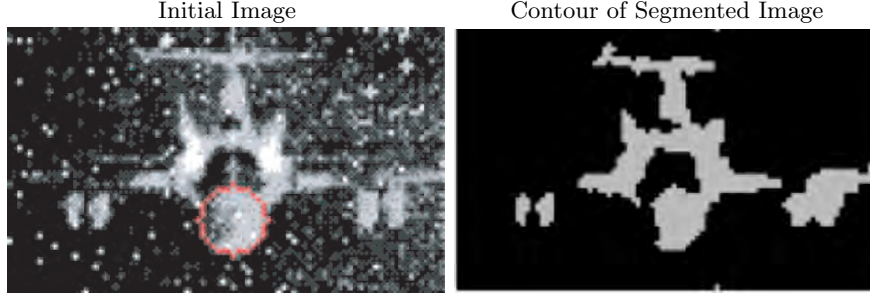


Fig. 2. The Chan–Vese algorithm is able to segment the image without edges.

$$F(c^+, c^-, \phi) = \mu |C(\phi)| + \lambda^+ \int_{\phi \geq 0} |u_0(x, y) - c^+|^2 dx dy \\ + \lambda^- \int_{\phi < 0} |u_0(x, y) - c^-|^2 dx dy.$$

Minimizing $F(c^+, c^-, \phi)$ with respect to the constants c^+ and c^- , for a fixed ϕ , yields the following expressions for c^+ and c^- as functions of ϕ :

$$c^+ = \text{average}(u_0) \text{ on } \phi \geq 0, \\ c^- = \text{average}(u_0) \text{ on } \phi < 0.$$

Minimizing the energy $F(c^+, c^-, \phi)$ with respect to ϕ , for fixed c^+ and c^- , using a gradient descent method, yields the associated Euler–Lagrange equation for ϕ , governed by the mean curvature and the error terms (see [11] for more details).

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda^+ (u_0 - c^+)^2 + \lambda^- (u_0 - c^-)^2 \right] \quad (3)$$

in Ω , and with the Neumann boundary conditions.

Using a level-set formulation with this model allows the initial contour to find any number of objects from an initial contour anywhere in the image. For general information, one may consult [44] and [51].

2.3 Piecewise Smooth Segmentation

Thus far, we have described objects that are assumed to have constant intensity. The piecewise smooth extension allows for two possible situations. One motivation is an algorithm that can simultaneously denoise and segment an image. A second situation occurs when an object’s intensity changes gradually. The general Mumford–Shah piecewise smooth functional [43] is defined as:

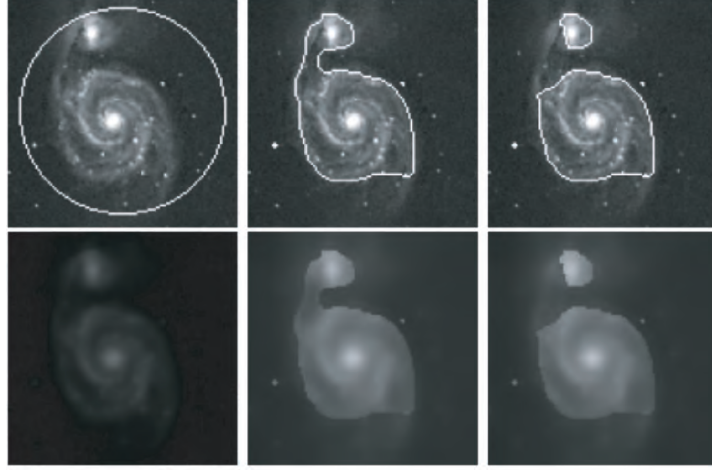


Fig. 3. A nebula is segmented using a single contour, giving intensity values, which are the same as in the original image.

$$\inf_{u, \Gamma} E_{ms}(u, \Gamma[u_0]) = \int_{\Omega} |u - u_0|^2 dx + \mu \int_{\Omega/\Gamma} |\nabla u|^2 dx + \nu |\Gamma|,$$

where μ and ν are positive constants. This allows for a varying intensity of the object, while keeping the boundaries constant. A two-phase function is defined with level set ϕ as follows:

$$u(x) = u^+(x)H(\phi(x)) + u^-(x)(1 - H(\phi(x))).$$

Here u^+ and u^- are C^1 functions up to the boundary at $\phi = 0$. Corresponding Euler–Lagrange equations are as follows:

$$\begin{aligned} \mu^2(u^+ - u_0) &= \Delta u^+ \text{ on } \phi > 0, & \frac{\partial u^+}{\partial n} &= 0 \text{ on } \phi = 0, \\ \mu^2(u^- - u_0) &= \Delta u^- \text{ on } \phi > 0, & \frac{\partial u^-}{\partial n} &= 0 \text{ on } \phi = 0. \end{aligned}$$

Denoising is done in the homogeneous region, while leaving the boundary $\phi = 0$ unchanged. In Figure 3 the correct features are captured within a single object segmentation.

2.4 Global Minima Via Convexification

The variational formulation in the Chan–Vese model is non-convex and a typical gradient-descent implementation is not guaranteed to converge to the global minimum and can get stuck in local minima. A typical case is where the contour is stuck at the outer boundary of an object with an interior hole. Various tricks can be devised to improve the global convergence. One technique,

which is used in the original paper [11], is to modify the delta function in the Euler–Lagrange equation so that it is nonzero everywhere. This corresponds to allowing contours to be initiated everywhere in the image, enhancing the chance of capturing the global minimum. Another idea is to initialize the optimization with a large number of small close contours uniformly distributed in the image, which has a similar effect.

A more novel, and fundamentally different, approach has been proposed more recently in [21, 22]. The basic idea is to convexify the objective function by taking advantage of the implicit geometric properties of the variational models. Using an auxiliary variable u , the Chan–Vese model can be recast in the following *convex* minimization problem:

$$\min_{c^-, c^+ \in \mathbb{R}} \min_{0 \leq u(x) \leq 1} \int_D |\nabla u| dx + \lambda \int_D [(c^+ - u_0)^2 - (c^- - u_0)^2] u(x) dx.$$

It is proved in [21, 22] that if $(c^+, c^-, u(x))$ is a solution of above minimization problem, then for almost every $\mu \in (0, 1)$, the triplet $(c^+, c^-, \chi_{x:u(x) \geq \mu}(x))$ is a global minimizer of the Chan–Vese model. Since the above minimization problem is convex, it admits many efficient implementations and thus this approach allows an efficient computation of the global minimization of the Chan–Vese model.

3 Multi-Channel Extensions in Chan–Vese Model

The Chan–Vese model described above is very flexible. This flexibility lends itself to expanding it in a variety of ways. Initially it was expanded to vector-valued systems. This allowed for combining multiple images simultaneously to segment the images and identify the key object. We introduce the generalized models below.

3.1 Vector-Valued Models

In this chapter, the Chan–Vese method [11] is extended to vector-valued images. An example of the vector-valued object detector can be seen in Figure 4. Each channel has a different piece missing, but when the two channels are combined, the complete object is detected. Another example where this algorithm is of particular interest is an occlusion occurring in one channel, while a second channel, complete yet noisier, is available. Another example is RGB images, where intensity detectors and channel-by-channel boundary detectors fail.

Let $u_{0,i}$ be the i th channel of an image on Ω , with $i = 1, \dots, N$ channels, and C the evolving curve. Each channel would contain the same image with some differences, for instance different wavelengths at which the image was taken, color images, etc. Let $\overline{c^+} = (c_1^+, \dots, c_N^+)$ and $\overline{c^-} = (c_1^-, \dots, c_N^-)$ be two unknown constant vectors.

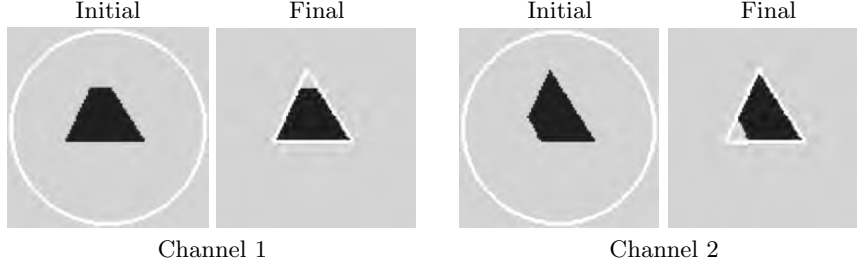


Fig. 4. Each channel has a different part of the same triangle missing. The vector-valued algorithm can detect the full triangle.

The extension of the Chan–Vese model to the vector case is:

$$F(\bar{c}^+, \bar{c}^-, \phi) = \mu \cdot \text{length}(C) + \int_{\text{inside}(C)} \frac{1}{N} \sum_{i=1}^N \lambda_i^+ |u_{0,i}(x, y) - c_i^+|^2 dx dy \\ + \int_{\text{outside}(C)} \frac{1}{N} \sum_{i=1}^N \lambda_i^- |u_{0,i}(x, y) - c_i^-|^2 dx dy,$$

where $\lambda_i^+ > 0$ and $\lambda_i^- > 0$ are parameters for each channel.

As in the scalar case, the model looks for the best vector-valued approximation taking only two values, the constant vectors $\bar{c}^+$ and $\bar{c}^-$. The active contour C is the boundary between these two regions. The energy balances the length of the contours in the image, with the fitting of u_0 , averaged over all channels. In this form, when the contour C surrounds the objects, our model can detect edges present in at least one of the channels, and not necessarily in all channels. We can associate this property with the syntax “OR”. Likewise we can imagine a system using the intersection of two objects. We will return to this boolean logic framework later in the paper.

An example can be found in multi-spectral images. In Figure 5, we have an airplane imaged from mid-wave and long-wave infrared channels. One channel is very noisy, making it very difficult to detect the edges of the entire airplane, while the other, less noisy, has a partial occlusion of the airplane. Each channel is insufficient for determination of the complete contour. However, in combination, most of the features are detected.

The vector-valued Chan–Vese model can also be used on color images. By dividing the image into red, green, and blue (RGB) channels, one can detect objects normally undetectable when the color image is transformed to a scalar intensity image. An example of this can be seen in Figure 6. We can see the “stop-light” in the RGB image, while the scalar intensity image has the bottom object missing. Channel-by-channel detection would also be insufficient in this case, since features of the image are not complete in any

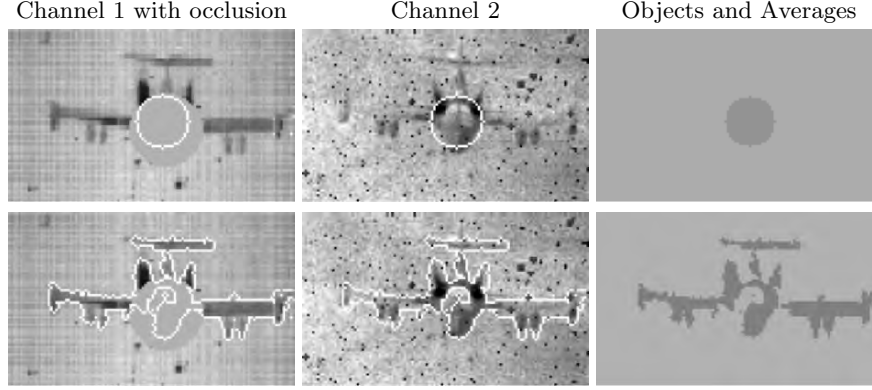


Fig. 5. While the first channel has little noise, but has an occlusion in it, the second channel is very noisy. From these two pictures, we try to detect as much of the airplane as possible. The parameters are as follows: $\mu = 0.001 \cdot 255^2$, $\lambda_1^+ = \lambda_1^- = 1$, $\lambda_2^+ = \lambda_2^- = 0.55$. In this example, we first performed a renormalization of the channels to $[0, 255]$.

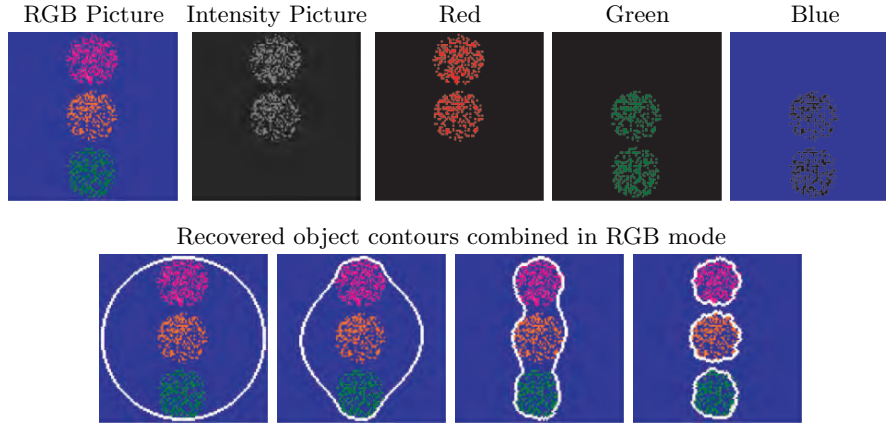


Fig. 6. We give here an example of a color image that has three objects of different colors, while the corresponding gray scale image only shows two of them. The boundary of all the circles is found, while in the gray-scale image the boundary of one of the circles would never be detected. Note that, since this image does not have gradient edges, a gradient-based algorithm would not be able to find the three objects. The parameters are as follows: $\mu = 0.06 \cdot 255^2$, $\lambda_i^+ = \lambda_i^- = 1$, for $i = 1, 2, 3$. (Color images in Figure A.13.)

single channel. Our model, however, detects all three features easily. Also note, in this particular example, the algorithm detects edges without gradient.

3.2 Texture Segmentation using Vector-Valued Models

There are several problems specific to texture segmentation. When the textures have the same intensities, it is very difficult for the standard segmentation models to tell them apart. Another problem inherent in textured segmentation is that it is often difficult to pick out the boundary between two textures because there is no sharp difference between them. Finally, any texture segmentation algorithm should be robust to noise, since texture has small patterns that are “noise”-like.

We do not assume any apriori knowledge or statistical information on the type of textures, or on the type of intensity, or on the location of boundaries. The proposed model described in detail in [10] is general, and can be applied in many situations.

For the texture discrimination, we propose to use Gabor [24] functions, having properties similar to those of early visual channels, being localized in space and frequency domains [23, 15]. The Gabor functions are convolved with the original textured image to obtain different channels. Some of these channels will be the input of the multi-channel active-contour algorithm.

For other possible transforms instead of the Gabor transform, for texture discrimination, such as wavelets; see for example [28]). This paper is related to many other works on active contours and texture segmentation, such as [46], (already mentioned above), [57, 56, 60, 52, 32]. Additional related papers are [37, 34, 6, 48]. Other related works on segmentation, edge-preserving smoothing, and vector-valued images (e.g., multi-channels, color, etc), are [16, 32, 50, 53, 54, 66].

Using all of the channels for segmentation is impractical. Some of the images are redundant while others add noise and obscure detection. At this point we divide our model into two parts: “supervised” texture segmentation, when the user chooses the “best” Gabor transforms, to be used as input channels; and “unsupervised” texture segmentation, where the Gabor transforms to be used are chosen by a dynamic automatic criterion.

The case of supervised texture segmentation allows one to use fewest number of transforms in order to segment the image, and as a result it does a very good job, with optimal computational efficiency. The case of unsupervised texture segmentation is similar to the work of [29, 63]. The criterion that we used for the automatic choice of the Gabor transforms is based on the following: we want the images to have the highest intensity differences relative to the mean of the image. Thus for each transformed channel i we calculate the following:

$$s_i = |c_i^+ - c_i^-|.$$

The s_i is calculated for each channel. Only n ($n < 45$) channels, corresponding to the first n largest values of s_i , are used in our active contour model as inputs, at the initial time. We recalculated the s_i at later iterations choosing the n largest values again. This allows for better choices of the channels as

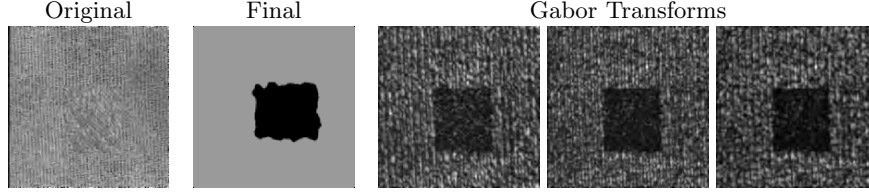


Fig. 7. Supervised model with three different Gabor transforms as input channels. Parameters: $\lambda_i = 1$, $\mu = 4000$, $\gamma_i = .3$. The boundary of the full square is found, and the binary segmentation is represented by “gray” and “black” (“black” if $\phi \geq 0$, and “gray” if $\phi < 0$).

the contour is refined closer to the desired object. This criterion does a good job of picking out automatically the “best” channels.

In Figure 7, there is a square in the middle of the image, but it is very hard to distinguish it. The Gabor transforms contrast the square, with the outside texture, and the active-contour model has no problem detecting the edges of the square. In Figures 8, we have used the unsupervised criteria for choosing the Gabor transforms. The segmentation is done well, with the criteria set for unsupervised segmentation.

3.3 Logic Operations on Region-Based Active Contours

The Chan–Vese method of active contours without edges is a region-based method. This is a significant benefit, as it is especially important when finding logical combinations of objects.

Rather than comparing contrast of the object, it compares the fitting errors of each channel. The model does not care that each channel has different intensity values, instead it wants a contour that will minimize the fitting errors based on the average value for each channel (Figure 9).

To set up the logical framework we define two separate logic variables, z_i^{in} and z_i^{out} , to denote whether a point (x, y) is in C or not:

$$z_i^{in}(u_0^i, x, y, C) = \begin{cases} 0, & \text{if } (x, y) \in C \text{ and } (x, y) \text{ inside the object in channel } i, \\ 1, & \text{otherwise;} \end{cases}$$

$$z_i^{out}(u_0^i, x, y, C) = \begin{cases} 1, & \text{if } (x, y) \notin C \text{ and } (x, y) \text{ is inside the object in channel } i, \\ 0, & \text{otherwise.} \end{cases}$$

A natural way to define z_i^{in} and z_i^{out} for the Chan–Vese model is as follows:

$$z_i^{in}(u_0^i, x, y, C) = \frac{|u_0^i(x, y) - c_+^i|^2}{\max_{(x, y) \in u_0^i} u_0^i},$$

$$z_i^{out}(u_0^i, x, y, C) = \frac{|u_0^i(x, y) - c_-^i|^2}{\max_{(x, y) \in u_0^i} u_0^i}. \quad (4)$$

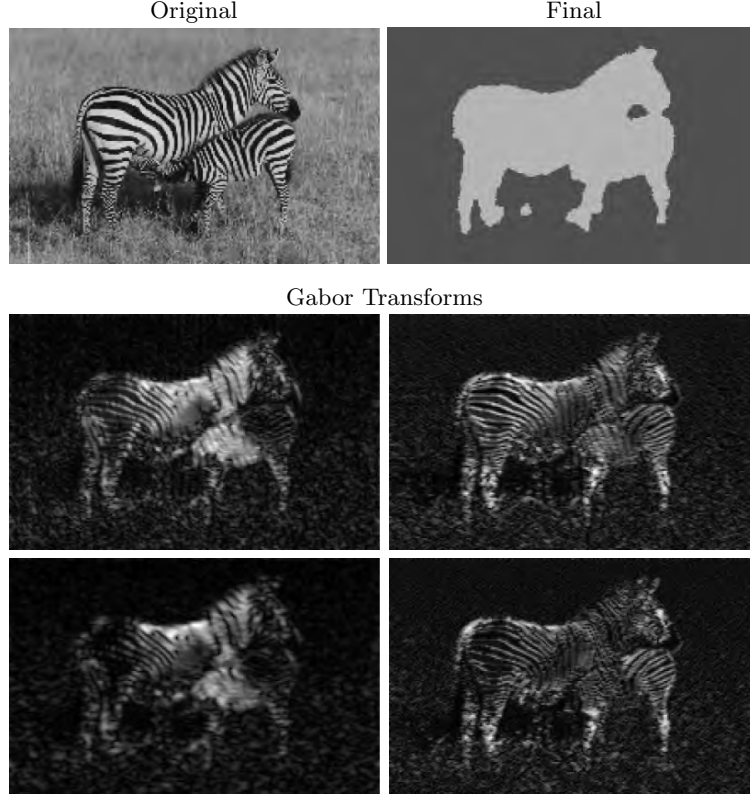


Fig. 8. Unsupervised texture segmentation with only four active transforms. It is successful in segmenting the zebras and disregarding the stripes.

Note that we use 0 as the “true” value, and 1 as the “false” value, which is the reverse of the usual convention. This is more convenient because our framework is based on a minimizing of an objective function and thus we want the 0 value to correspond to “true”.

For the complement of the object in channel i we define:

$$\begin{aligned} z_i^{in'} &= 1 - z_i^{in} \\ z_i^{out'} &= 1 - z_i^{out} \end{aligned} \tag{5}$$

Following the structure of logic operators, we now want to define a truth table for the logic model the the variables described above. We treat the points inside C separately from those outside C .

Continuing with the two-channel example $A_1 \cup A_2$, we define it in truth table form. The truth table needs to reflect the union of z_i^{in} and the intersection of z_i^{out} . For the point $(x, y) \in C$ the variable z_i^{in} is defined. If the

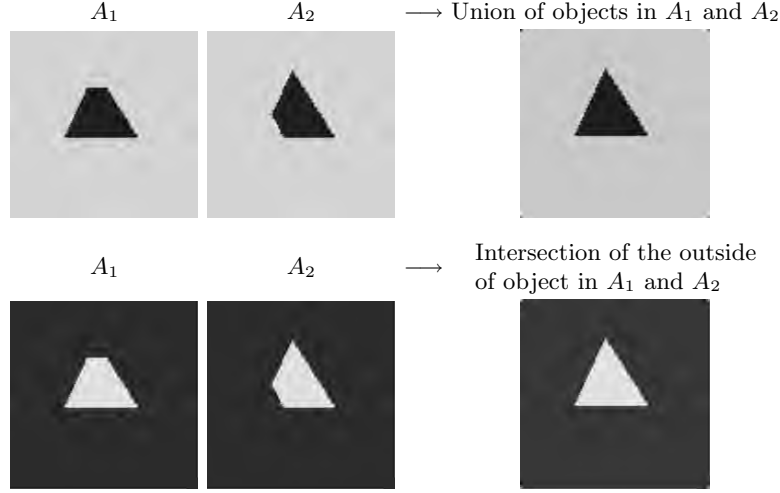


Fig. 9. Logic operations inside and outside the object. The upper triple of images shows that the union of the inside (black) region gives the union of the 2 objects in A_1 and A_2 . The bottom triple shows that the intersection of the outside (black) region gives the complement to the union of two objects.

point $(x, y) \in C$ is in the object in either channel, the logic model returns 0, otherwise it returns 1—this reflects the union of the inside of the object. If $(x, y) \in \Omega \setminus C$, the variable z_i^{out} is defined. The logic model returns 0 if (x, y) is *not* in the object in either channel, otherwise it will return 1, - this represents the intersection of the outside of the object. The column marked $A_1 \cup A_2$ relates this information. The logic operations $A_1 \cap A_2$ and $A_1 \cap \neg A_2$ are calculated in a similar fashion. For intersection of objects, we take the intersection of the inside of objects and the union of the outside of objects. For negation we substitute z'_i for z_i as shown in (5).

For the union and intersection function of logic variables we choose:

$$f_{z_1 \cup z_2} = (z_1 \cdot z_2)^{\frac{1}{2}}, \quad f_{z_1 \cap z_2} = 1 - ((1 - z_1)(1 - z_2))^{\frac{1}{2}}.$$

The square roots of the products are taken to keep them of the same order as the original scalar model. Combining the interpolation functions for union of inside the objects, and intersection outside the objects we get the union of objects:

$$f_{A_1 \cup A_2}(x, y) = \sqrt{z_1^{in}(x, y)z_2^{in}(x, y)} + 1 - \sqrt{(1 - z_1^{out}(x, y))(1 - z_2^{out}(x, y))}.$$

Likewise, to get the intersection of objects, we combine the intersection of the inside with the union of the outside, resulting in the following objective function for the intersection of objects:

$$f_{A_1 \cap A_2}(x, y) = 1 - \sqrt{(1 - z_1^{in}(x, y))(1 - z_2^{in}(x, y))} + \sqrt{z_1^{out}(x, y)z_2^{out}(x, y)}.$$

In the above, we have used the interpolation functions to directly derive the objective functions corresponding to a given logical expression. Even though we have by-passed the corresponding truth table, it can be easily verified that the resulting objection functions do interpolate the function values given in the truth table. The functional may be written using the level-set formulation as described in Section 2. Now we can rewrite the functional F for a general $f(z_1^{in}, z_1^{out}, \dots)$ using the level-set function ϕ .

The objective function for the variational model is:

$$F(\phi, c^+, c^-) = \mu|C(\phi)| + \lambda \left[\int_{\Omega} f_{in}(z_1^{in}, \dots, z_n^{in})H(\phi) + f_{out}(z_1^{out}, \dots, z_n^{out})(1 - H(\phi))dx \right].$$

Derivation of the Euler–Lagrange equation is similar to that of the scalar model and yields the following differential equation (with Neumann boundary conditions):

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda (f_{in}(z_1^{in}, \dots, z_n^{in}) - f_{out}(z_1^{out}, \dots, z_n^{out})) \right],$$

which at steady state gives the solution.

For example, for the two logic models presented earlier, the corresponding Euler–Lagrange equations are:

$$\begin{aligned} \frac{\partial \phi_{L_1(A_1) \cup \dots \cup L_n(A_n)}}{\partial t} &= \\ \delta_{\epsilon}(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda \left(\left(\prod_{i=1}^n l_i(z_i^{in}) \right)^{\frac{1}{n}} + 1 - \left(\prod_{i=1}^n (1 - l_i(z_i^{out})) \right)^{\frac{1}{n}} \right) \right], \\ \frac{\partial \phi_{L_1(A_1) \cap \dots \cap L_n(A_n)}}{\partial t} &= \\ \delta_{\epsilon}(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda \left(1 - \left(\prod_{i=1}^n (1 - l_i(z_i^{in})) \right)^{\frac{1}{n}} + \left(\prod_{i=1}^n l_i(z_i^{out}) \right)^{\frac{1}{n}} \right) \right]. \end{aligned}$$

Even though the form is complicated, the implementation is very similar to that of the scalar model that is in (3). The details for this scheme can be found in [11, 49]. In this section, we show some examples of the performance of the logical active-contour models described in Section 3.

We show a real life example in Figure 10 with two brain images. They are two MRIs of the brain taken in a time sequence, each with a synthetic tumor placed in a different spot. Using logic operation $A_1 \cap \neg A_2$, the tumor in the first image may be extracted, i.e., the logic operations find the object in the first image that is different from the second. The reverse is also true. Using the logic model that describes $\neg A_1 \cap A_2$, the model finds the object in the

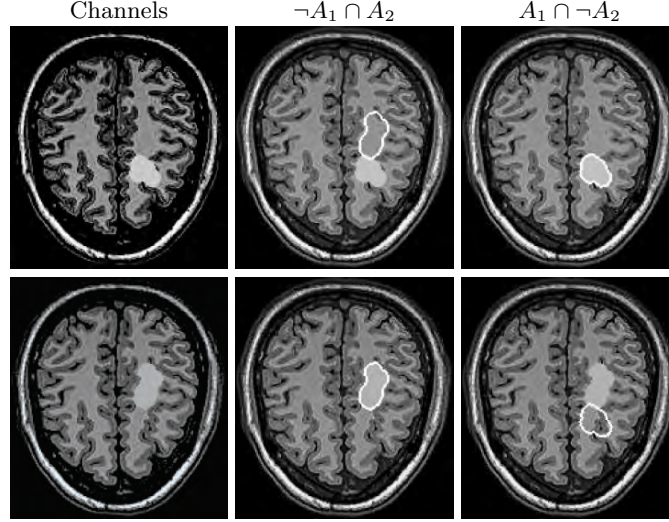


Fig. 10. Region-based logic model on a MRI scan of the brain. The first channel A_1 , has a synthetic brain tumor in one place; in the second image the synthetic brain tumor is in a different place. The images are registered. By design we want to find the tumor that is in A_1 and not A_2 , $A_1 \cap \neg A_2$. Likewise we want to find the tumor in A_2 that is not in A_1 and $\neg A_1 \cap A_2$.

second image that is not in the first. This happens to be a very complicated example as there are a lot of features and textures. Not only does the model find the tumor, but using logic operations gives the user the capability to define more precisely how information from the different channels are to be combined in order to obtain a desired segmentation, as well as the freedom to use all possible logical combinations using a systematic framework.

In practical terms, the logic framework allows for a single solution global minimum as the union or intersection of the object depending on the model chosen. The vector-valued function depends on the initial contour for the final output, giving either union or intersection of the objects.

3.4 Target Tracking in Video

In this section, we show how the Chan–Vese segmentation model can be extended to track deforming objects in video sequences. This methodology was developed by Moelich [40, 41]. Since the Chan–Vese algorithm finds an optimal piecewise-constant approximation to an image, this algorithm works best in tracking objects that have nearly uniform intensity. The main idea is to sequentially segment the frames of a video sequence by using the final partition from one frame as the initial partition of the next.



Fig. 11. Results of tracking an object using a modified version of the Chan–Vese algorithm. (Color images in Figure A.14.)

An estimate of each initial contour, which is based on a number of previous frames, can also be used. This, however, is not necessary unless the frame-to-frame motion is large compared to the size of the object. Figure 11 shows sample frames from the complete sequence. Note that the algorithm is able to capture much information about the person being tracked, including gait and posture.

Some important modifications are made to the basic Chan–Vese model to adapt it to tracking objects. The first is to use a local background model, where the background is isolated to the region outside, but close to the contour. Second, reinitializing the distance function maintains a local minima. Furthermore, once the desired object is identified, the segmentation should occur in the region of interest surrounding the object to maintain a “global” minima.

This method can fail when the estimated position of the object in a frame is far from its true position. This can happen when the frame-to-frame motion of the object is large relative to the size of the object. In this case, the segmentation contour is not in contact with the object and can either begin to isolate a similar nearby object, or vanish. Little can be done if it begins to segment another similar object. If the contour vanishes, however, it can be successively enlarged until it finds the desired object. The image in Figure 12 is the completed segmentation of a frame. This contour is used as the initial contour of the next frame. Due to camera motion, the contour misses the object in the second frame. Since the estimated intensity for the object is not within the contour, the contour shrinks until it vanishes. When this happens, the algorithm successively enlarges the contour until it contacts the object, in which case the algorithm is able to isolate it.

The use of the level-set framework makes “enlarging” the segmentation contour trivial. Recall that the segmentation contour is implicitly defined as the zero level set of a higher-dimensional function φ , where $\varphi < 0$ inside the contour and $\varphi > 0$ outside of the contour. Enlarging the segmentation contour is the same as “lowering” the level set φ . By continually reinitializing the distance function, the value of $|\nabla\varphi|$ is approximately equal 1 near the contour (zero level set). To increase the size of the contour by a fractional amount f , we can simply modify φ as follows:



Fig. 12. Illustration of how algorithm handles position errors. The child moved far from frame to frame, by enlarging the contour the child is found in the following image. (Color images in Figure A.15.)



Fig. 13. Tracking in presence of background clutter and poor contrast. (Color images in Figure A.16.)

$$\varphi^{\text{new}} = \varphi^{\text{old}} - f d/2, \quad (6)$$

where d is an estimate of the diameter of the contour, which is made before the segmentation is applied. We used the larger of the height and width of the contour in the image as the estimate of the diameter.

Figure 13 gives an example sequence that was produced by this algorithm, tracking the car successfully in a highly cluttered environment.

3.5 Color Segmentation

The Chan–Vese segmentation model was originally developed to segment piecewise constant images. This algorithm was extended to isolate regions of constant color with the vector-valued models. We described a method for isolating objects that are composed of more than one color. This methodology was developed by Moelich [42].

An object of interest is often composed of a small number of different colors. For example, the cat in Figure 14 is composed of the colors black and white. A piecewise constant segmentation algorithm that is based on intensity alone, would not be able to isolate the entire cat as a single object.

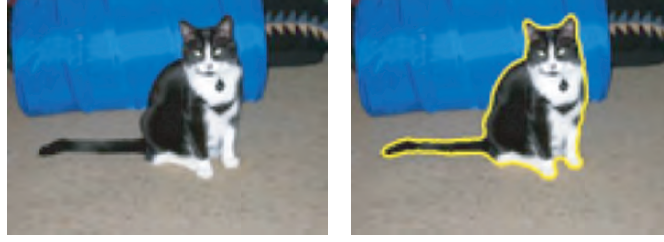


Fig. 14. A black and white cat and output of color logic model. (Color images in Figure A.17.)

This segmentation model assumes some apriori knowledge about the colors of the object to be isolated. This can be introduced to the algorithm, for example, by “clicking” on the desired colors in a graphical display. Given a color image $I: \Omega \rightarrow \mathbb{R}^3$ and a set of colors $c = (c_1, \dots, c_n)$, the prototype color logic model uses OR and AND framework described in the previous section to find a contour C that minimizes the energy

$$E(C; c) = \lambda_{in} \int_{\Omega_{in}} F_{in}(I(x); c) + \lambda_{out} \int_{\Omega_{out}} F_{out}(I(x); c) + \mu \text{length}(C), \quad (7)$$

where

$$F_{in}(I(x); c) = \left(\prod_{i=1}^n k_i \|I(x) - c_i\| \right)^{1/n}, \quad (8)$$

$$F_{out}(I(x); c) = 1 - \left(\prod_{i=1}^n k_i \|I(x) - c_i\| \right)^{1/n}, \quad (9)$$

and where λ_{in} , λ_{out} , and μ are design parameters, and Ω_{in} and Ω_{out} are the regions interior and exterior to the contour C , respectively. The values of k_i are chosen to normalize the quantities $\|I(x) - c_i\|$ and to ensure that they lie in the unit interval.

Figure 15 shows three additional segmentations that were produced by this model. In each case, two to six different colors were chosen before the segmentation.

The models for the two regions, given by (8) and (9) above, are effective for many images. When the colors in the background are similar to the colors in the object, then an independent background model can be used. In this case, the model for the background in (9) is replaced by

$$F_{out}(I(x); c^{out}) = \left(\prod_{j=1}^m k_j \|I(x) - c_{out,j}\| \right)^{1/m}, \quad (10)$$



Fig. 15. Additional example of color logic model. (Color images in Figure A.18.)



Fig. 16. Illustration of improved background model. Choosing three colors (left) or two colors (right) with first background model, and choosing three object colors and three background colors for improved background model. (Color images in Figure A.19.)

where c_{out} is the set of m colors used to describe the exterior region. The two images on the left of Figure 16 were generated without this new model. In each of these cases, the segmentation was stopped before completion. In the image on the left, the colors red, white, and blue were selected from the flag. Since the color of the clouds behind the flag are nearly white, the algorithm considers them part of the object, and the segmentation contour grows to include them. In the middle image, only the colors red and blue were chosen. In this case the clouds, along with the white strips are excluded. Because of the regularity term, the thin red strips are also excluded.

The improved background model (10) was applied to the image on the right of Figure 16. In this case, the colors red, white, and blue were selected from the flag to define the object model, and additional three colors were selected from the background to define the background region. The use of independent models for the object and background regions provides the desired segmentation.

3.6 Image Registration

An algorithm for the joint segmentation and registration of images is described. Similar to other algorithms that we have discussed, the main idea is to use information from more than one image to develop a segmentation. We do not assume that the images are registered, or “lined up.” This algorithm simultaneously finds both the segmentation and the registration between the

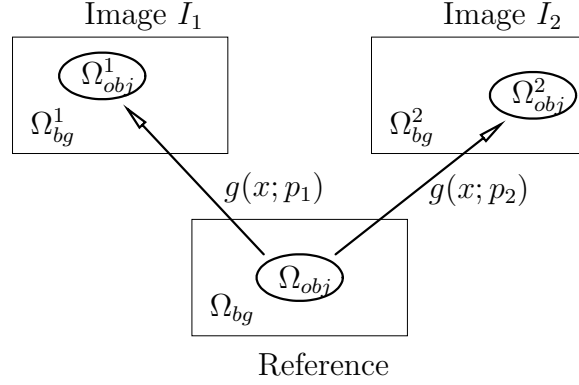


Fig. 17. Individual contours are mappings of a reference contours.

images. This approach was created by Yezzi, Zöllei, and Kapur [64], and further developed by Moelich [41]. Our description focuses on the case of two images; however, the same method can be applied to a larger number of images.

Consider two, possibly unregistered images, $I^1: \Omega \rightarrow \mathbb{R}$ and $I^2: \Omega \rightarrow \mathbb{R}$. Let $\{\Omega_{obj}^1, \Omega_{bg}^1\}$ denote the segmentation of the image I^1 and let $\{\Omega_{obj}^2, \Omega_{bg}^2\}$ denote the segmentation of image I^2 . These two segmentations are viewed as the mapping of a reference segmentation $\{\Omega_{obj}, \Omega_{bg}\}$ under a parameterized mapping g . Figure 17 illustrates this relationship, where p_1 and p_2 represent two sets of parameters for the mapping g .

The segmentation and registration $p = (p_1, p_2)$ are found by minimizing a segmentation energy that is defined as

$$E(\Omega_{obj}, \Omega_{bg}; p) = E_{obj}(\Omega_{obj}; p) + E_{bg}(\Omega_{bg}; p) + \mu |\partial \Omega_{obj}|, \quad (11)$$

where

$$E_{obj}(\Omega_{obj}; p) = \int_{\Omega_{obj}} \lambda_{obj} F_{obj}(x; p) dx, \quad (12)$$

$$E_{bg}(\Omega_{bg}; p) = \int_{\Omega_{bg}} \lambda_{bg} F_{bg}(x; p) dx. \quad (13)$$

The expressions for the region descriptors $F_{obj}(x; p)$ and $F_{bg}(x; p)$ depend upon which segmentation models are used.

There are many valid choices for the mapping g , but for many applications a simple Euclidean transformation $g(x; p) = MRx + T$ is adequate, where

$$M = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix}, \quad R = \begin{bmatrix} \cos \Delta\theta & -\sin \Delta\theta \\ \sin \Delta\theta & \cos \Delta\theta \end{bmatrix}, \quad \text{and} \quad T = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

The parameters of the transformation are given by $p = (\Delta x, \Delta y, \Delta\theta, m)$, where Δx and Δy are translation, $\Delta\theta$ is rotation about the center, and m is

magnification. When $p = (0, 0, 0, 1)$, the transformation $g(\cdot; p)$ is the identity map.

The segmentation energy (11) depends on both the partition of the image and on the registration parameters. This energy can be minimized by “inter-leaving” the processes of segmentation and registration, as suggested in [64]. During each iteration of the algorithm, the segmentation, i.e., the level-set function φ , is first held constant while the estimates of the registration parameters are improved, then these parameters are held fixed while the level-set function φ is evolved.

The registration parameters are improved by using a gradient-descent line search. The direction of the search is found by taking numerical derivatives of the energy $E^{obj}(\Omega_{obj}; p)$ with respect to the components of $p = (p_1, p_2)$. Since p_1 and p_2 are independent, it is useful to update each set of parameters separately. Also, since translation, rotation, and magnification have different scales, it is useful to have different time steps for these components.

The segmentation is improved by making a small evolution of the level-set function by numerically integrating

$$\frac{\partial \varphi}{\partial t} = \delta_\varepsilon(\varphi) \left(\lambda_{obj} F_{obj}(x; p) - \lambda_{bg} F_{bg}(x; p) + \mu \operatorname{div} \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) \right), \quad \text{in } \Omega, \quad (14)$$

$$\frac{\partial \varphi}{\partial n} = 0, \quad \text{on } \partial \Omega, \quad (15)$$

$$\varphi(x, 0) = \varphi_k, \quad \text{in } \Omega \quad (16)$$

for a few steps. The energy decreases with each iteration. The process of alternatively improving the registration and segmentation continues until the algorithm converges.

When the initial estimate of the registration parameters are poor, an initial registration phase can be used to put the segmentation contours in contact with the object of interest in each image. This initial registration phase assumes that an apriori estimate of the average intensities of the object of interest is known. The initial phase can be applied to either, or both images. For sake of discussion, we assume that the initial guess for p_1 is known to be reasonable, but that the error in the initial guess for p_2 can be large. It is further assumed, in this case, that the average intensity of Ω_{obj}^1 is an estimate for $\hat{c}_{obj}^2$, of the intensity of Ω_{obj}^2 .

The estimate $\hat{c}_{obj}^2$ is used to construct an initial registration phase energy

$$E_\psi(C_2) = \frac{1}{m} \int_{C_2} \psi^m(x) ds, \quad (17)$$

where $\psi(x)$ is the distance from x to the intensity of interest in I_2 and where $C_2 = \partial \Omega_{obj}^2$ is the segmentation contour. The value of m can be taken as either 1 or 2. A value of $m = 1$ usually gives a faster convergence, although using $m = 2$ gives better behavior near the minima.



Fig. 18. Typical behavior of the algorithm. Initial contour (top), end of initial registration phase (middle), and final segmentation (bottom). (Color images in Figure A.21.)

A gradient descent is used to minimize the energy E_ψ of the initial registration phase. The values of Δx and Δy , which are registration parameters for the image, are updated using the following equations:

$$\begin{aligned} \frac{\partial \Delta x}{\partial t} &= -\frac{\partial E_\psi}{\partial x} = -\int_{C_2} \psi^{m-1} \frac{\partial \psi(\mathbf{x})}{\partial x} ds, \\ \frac{\partial \Delta y}{\partial t} &= -\frac{\partial E_\psi}{\partial y} = -\int_{C_2} \psi^{m-1} \frac{\partial \psi(\mathbf{x})}{\partial y} ds. \end{aligned} \quad (18)$$

Figure 18 illustrates the typical behavior of the complete algorithm. In this example, the piecewise constant Chan–Vese segmentation model is used. The images in the left and right columns, respectively, were taken from slightly different perspectives, at slightly different times. The estimates of the registration parameters are reasonable for the image on the left, but not for the image on the right. The initial registration phase is used to drive the contour in the image on the right toward the object of interest. Once the initial phase energy is minimized, the phase changes and joint segmentation and registration is used to both isolate the object and determine the registration parameters.

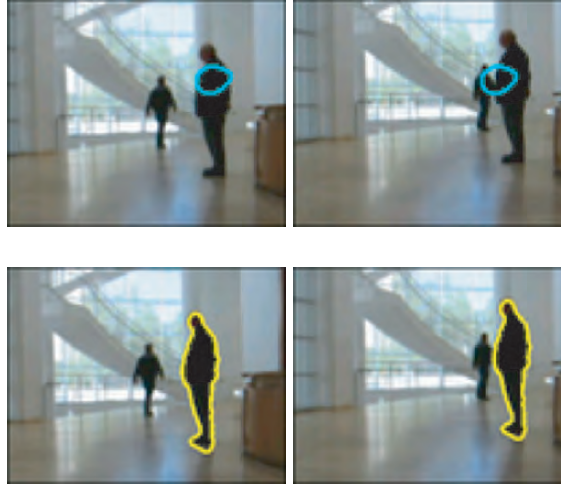


Fig. 19. Logical AND model restricts the segmentation. Initial contour (top) and logical AND (bottom). (Color images in Figure A.22.)

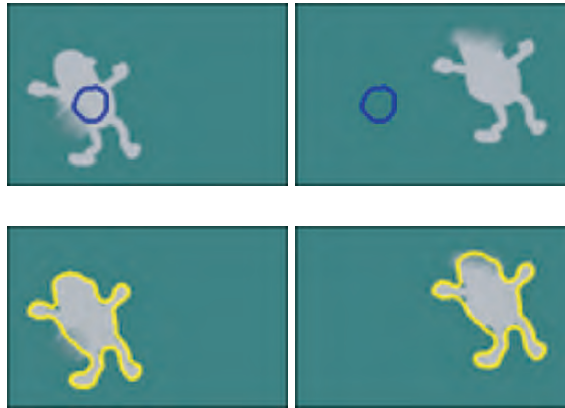


Fig. 20. Logical OR model combines information. Initial contour (top) and final segmentation (bottom).

Figure 19 illustrates how the logical AND model can be used. The image of the person on the left is used as a template to restrict the segmentation of the image of the person on the right. The initial contours are shown in the top row, and the final contours are shown on the bottom. Note the the person in the background is ignored.

In Figure 20, the logical OR model is used to reconstruct an image from two incomplete images.

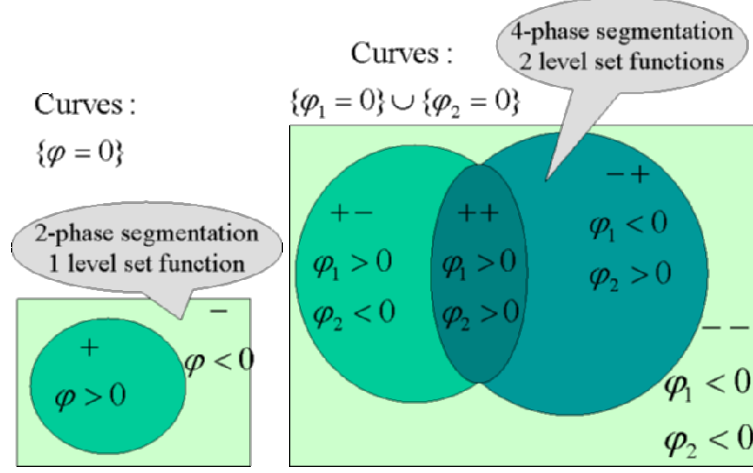


Fig. 21. A physical representation of the difference in region segmentation between one and two level sets. The number of regions possible is 2^m where m is the number of level sets, here $m = 1, 2$.

4 Multi-Phase Extensions

Several multi-phase algorithm extensions are described below. The original one developed by Chan and Vese [61] adds a new contour to add new regions. For m contours one would be able to develop algorithms for 2^m regions (Figure 21). This can be bulky to program. Further work has been done on multiphase methods to increase their efficiency and ease of programming. A recursive method was introduced by [25] that segments the image in a hierarchical way. First into two regions, then segmenting each region into two new regions, and so on. Piecewise constant method by [36] motivated by island dynamics for modeling epitaxial growth is used for memory efficiency. A multi-layer method by [13] uses different ranges of a function for different regions, however, nested regions and triple junctions require more than one function. Binary methods were introduced by [35] and [55], which require no Delta or Heaviside functions, obtain direct optimization, for faster implementation.

4.1 Multi-Phase Active Contours without edges

In the previous sections we have discussed segmentation for a single object. We now show the multi-phase extensions that have been suggested by [61]. The initial multi-phase algorithm follows the natural extension of the piecewise constant Mumford-Shah functional, as shown below:

$$\inf_{u, \Gamma} E_{ms}[u, \Gamma, u_0] = \sum_{i=1}^N \int_{\Omega_i} |u_0 - c_i|^2 + \mu |\Gamma|,$$

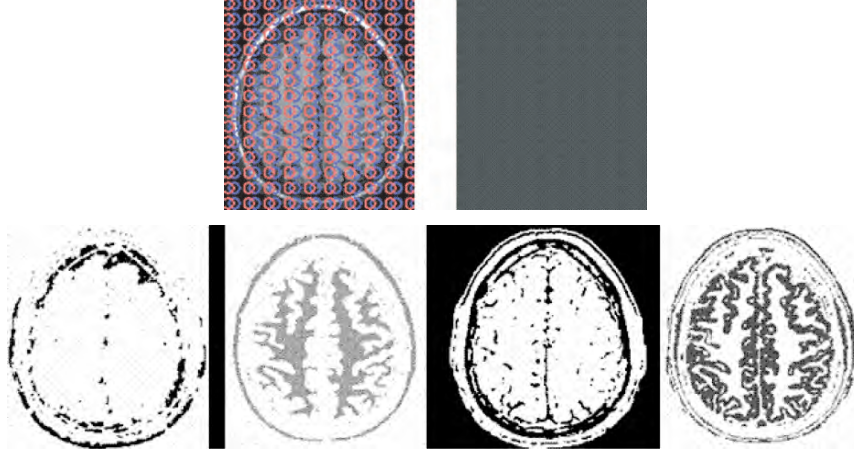


Fig. 22. The image of the brain with initial contours are in section (upper row) and the final output split into 4 different regions (lower row).

where Γ'_i s are the connected components of $\Omega \setminus \Gamma$ and $u = c_i$ on Γ_i . Rewriting this in level-set form, we see that for m level sets there are $n = 2^m$ phases that partition the image into n regions

$$u = c_{11}H(\phi_1)H(\phi_2) + c_{12}H(\phi_1)(1 - H(\phi_2)) \\ + c_{21}(1 - H(\phi_1))H(\phi_2) + c_{22}(1 - H(\phi_1))(1 - H(\phi_2)).$$

The Mumford–Shah segmentation becomes:

$$E_4[c, \Phi | u_0] = \int_{\Omega} |u_0(x) - c_{11}|^2 H(\phi_1)H(\phi_2) dx \\ + \int_{\Omega} |u_0(x) - c_{12}|^2 H(\phi_1)(1 - H(\phi_2)) dx \\ + \int_{\Omega} |u_0(x) - c_{21}|^2 (1 - H(\phi_1))H(\phi_2) dx \\ + \int_{\Omega} |u_0(x) - c_{22}|^2 (1 - H(\phi_1))(1 - H(\phi_2)) dx \\ + \mu \int_{\Omega} |\nabla H(\phi_1)| + |\nabla H(\phi_2)| dx.$$

Minimizing the Mumford–Shah equation leads to the Euler–Lagrange equation, fixing Φ and minimizing c , then the reverse:

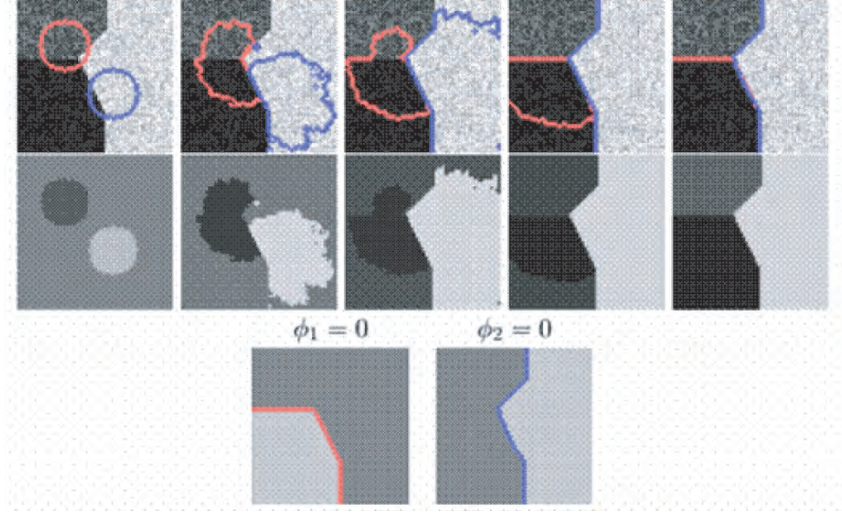


Fig. 23. A synthetic image with a noisy t-junction is segmented using two level sets [61].

$$\begin{aligned}
 c_{ij}(t) &= \text{average of } u_0 \text{ on } (2i-1)\phi_1 > 0, \quad (2j-1)\phi_2 > 0, \quad i, j = 1, 2 \\
 \frac{\partial \phi_1}{\partial t} &= \delta(\phi_1) \left[\mu \nabla \left(\frac{\nabla \phi_1}{|\nabla \phi_1|} \right) - (|u_0 - c_{11}|^2 - (u_0 - c_{12})^2) H(\phi_2) \right. \\
 &\quad \left. - ((u_0 - c_{21})^2 - (u_0 - c_{22})^2) (1 - H(\phi_2)) \right], \\
 \frac{\partial \phi_2}{\partial t} &= \delta(\phi_2) \left[\mu \nabla \left(\frac{\nabla \phi_2}{|\nabla \phi_2|} \right) - (|u_0 - c_{11}|^2 - (u_0 - c_{12})^2) H(\phi_1) \right. \\
 &\quad \left. - ((u_0 - c_{21})^2 - (u_0 - c_{22})^2) (1 - H(\phi_1)) \right].
 \end{aligned}$$

The equations are effected by mean curvatures and jumps of data energy terms across the boundary. We show two examples in Figure 22, and t-junction example shows the robustness of the methods in Figure 23, the equations for which can be found in [61].

4.2 Piecewise Constant Level-Set Method (PCLSM)

The motivation of this model is the same as the one shown above, but to accomplish this in a single level set. The multi-region segmentation model is defined using a single function ϕ which is a piecewise constant function taking the values:

$$\phi = i \text{ in } \Omega_i, i = 1, 2, \dots, n.$$

The discontinuities of ϕ give curves that separate the regions [36]. Using this definition of regions the minimization problem for image u_0 is:

$$\min_{\mathbf{c}, \phi, K(\phi)=0} F(\mathbf{c}, \phi) = \int_{\Omega} |u - u_0|^2 dx + \beta \sum_{i=1}^n \int_{\Omega} |\nabla \psi_i| dx,$$

where the function ψ_i and the constraint are:

$$\psi_i = \frac{1}{\alpha_i} \Pi_{k=1, k \neq i}^n (\phi - k) \text{ and } \alpha_i = \Pi_{k=1, k \neq i}^n (i - k),$$

$$K(\phi) = \Pi_{i=1}^n (\phi - i)$$

and u is defined by

$$u = \sum_{i=1}^n c_i \psi_i.$$

For details on calculating the minimum see [36, 12]. Updating the constant values is very ill-posed, a small perturbation in ϕ can yield a large jump in $\mathbf{c}$, putting some constraints. The benefit of this algorithm is that it can segment very noisy images, as can be seen in Figure 24. Even though the star is very noisy, PCLSM is able to segment the image.

Further work has been done that minimizes only the level-set function, not the constant values, and both gradient-descent and Newton's method are used to solve the Euler-Lagrange differential equations [58]. An example is shown for a two-phase image segmentation. A landscape that has some complicated shapes is segmented using both Newton's method and gradient-descent method in Figure 25.

4.3 Multi-Layer Active Contours without Edges

The multi-layer method uses a single ϕ with layers. The idea was inspired by multilayer techniques for modeling epitaxial growth [5]. The minimization described is non-convex, non-unique, and works locally, but the implementation is simple and the results are good.

Below we show the energy equation for a single function ϕ with m levels $l_1 < l_2 < \dots < l_m$. This will split the image into $m + 1$ regions with the following boundaries:

$$R_m = x \in \Omega; l_{m-1} < \phi(x) < l_m$$

The energy functional for this layering is as follows:

$$\begin{aligned} \inf_{c_1, \dots, c_{m+1}, \phi} F(c_1, c_2, \dots, c_{m+1}, \phi) &= \int_{\Omega} |f(x) - c_1|^2 H(l_1 - \phi(x)) dx \\ &+ \sum_{i=2}^m \int_{\Omega} |f(x) - c_i|^2 H(\phi(x) - l_i) dx \\ &+ \int_{\Omega} |f(x) - c_{m+1}|^2 H(\phi(x) - l_m) dx \\ &+ \mu \sum_{i=1}^m \int_{\Omega} |\nabla H(\phi - l_i)| dx. \end{aligned}$$

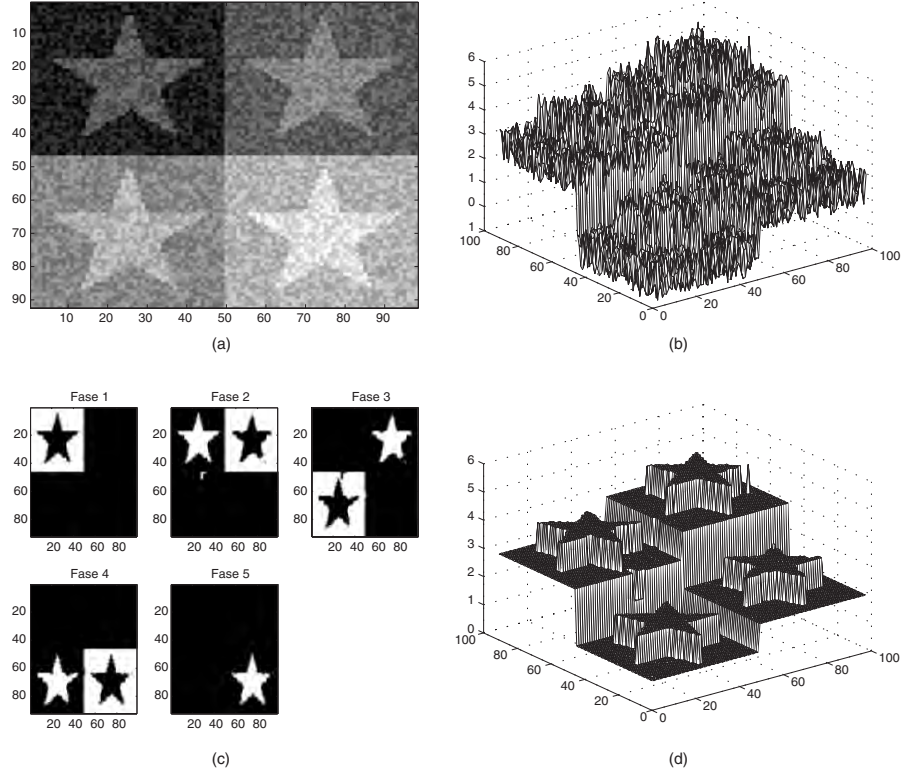


Fig. 24. (a) Observed image u_0 (SNR about 10.6). (b) Initial level set ϕ , (c) Different phases using PCLSM where $\phi = 1 \vee 2 \vee 3 \vee 4$ are depicted as bright regions. (d) View of ϕ at convergence. for further details see [36].

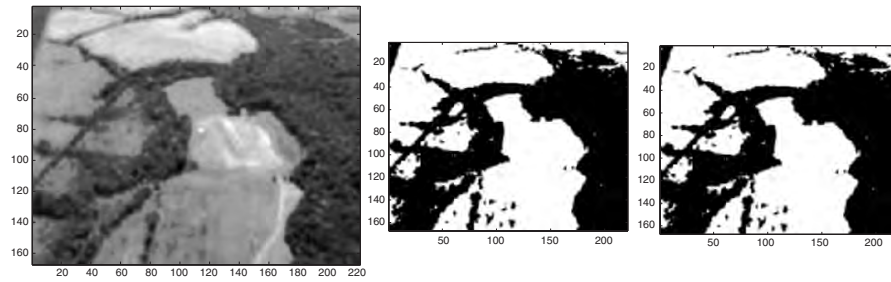


Fig. 25. From left to right: observed image, segmentation using Newton's method, and segmentation using gradient descent.

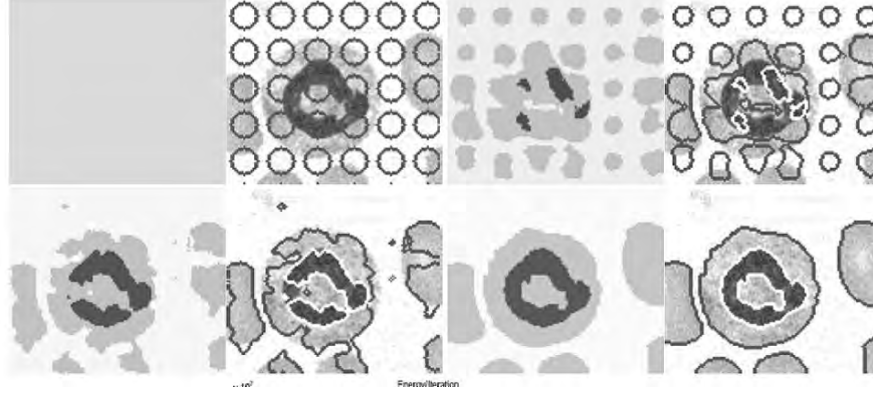


Fig. 26. Segmentation of a noisy real blood cells image using one level-set function and two levels, for further details see [13].

The Euler–Lagrange equations are as follows:

$$\begin{aligned}
 c_1(t) &= \frac{\int_{\Omega} f(x) H(l_1 - \phi(x, t)) dx}{\int_{\Omega} H(l_1 - \phi(x, t)) dx}, \\
 c_i(t) &= \frac{\int_{\Omega} f(x) H(\phi(x, t) - l_{i-1}) H(l_i - \phi(x, t)) dx}{\int_{\Omega} H(\phi(x, t) - l_{i-1}) H(l_i - \phi(x, t)) dx}, \\
 c_{m+1}(t) &= \frac{\int_{\Omega} |f(x) - c_1|^2 H(l_1 - \phi(x, t)) dx}{\int_{\Omega} H(\phi(x, t) - l_m) dx}.
 \end{aligned}$$

For further algorithmic development see [13]. In Figure 26 a noisy image of a red blood cell is segmented.

5 Fast Algorithms

The image processing techniques described above are very promising, but they could be somewhat slow even on simple images, because the model iterates until it comes to a stable solution. Ways to speed up the algorithms have therefore been discussed in a number of papers.

5.1 Direct Optimization

One solution by [44] is to solve the partial differential equation in a narrow band, close to where the level set is zero. Another possibility proposed by [61] is to simply use implicit methods and take large steps. Multigrid methods have been developed [59]. New ideas that have been developed over the last

several years include operator splitting by [26], direct optimization [19, 55], and threshold dynamics.

One approach that has been developed is to use the level-set function, without solving any differential equations. For problems that are formulated using level sets ϕ and can be written in the form:

$$\min_{\phi} F(H(\phi)),$$

the values of the objective function F are calculated directly. F does not need to be differentiable, which allows an extra degree of freedom in picking a model. The values of the level set is not needed, just the sign. Instead of evolving the differential equation, one can calculate the original objective function, then note the changes to the objective function if the sign of the level-set function is changed for the particular pixel. The algorithm follows three straightforward steps. It is initialized and objective function F is calculated for the initial partition of $\phi > 0$ and $\phi < 0$. For each point, x in the image, if the energy F decreases, then change $\phi(x)$ to $-\phi(x)$. Continuing to recalculate F through the image until the energy F remains unchanged.

The requirements of this algorithm are satisfied by the Chan–Vese model. The algorithm for the Chan–Vese model follows the three-step process described above. When a local change to $\phi(x)$ occurs, the global values of the energy can be changed with a local calculation. For two-phase images it is proven in [55] that this algorithm converges in one sweep independently of the sweep order. It was further proven by [20] that this holds for images with small noise. In Figure 27, the convergence occurs in four steps.

5.2 Operator Splitting

Another fast method that was developed by Gabou and Fedkiw [26] also uses only the sign of the level-set function rather than the value. It splits the curvature from the data-fidelity term. First, it calculates the Euler–Lagrange equation without the length term. This allows the method to take large time steps. The length term is handled by a separate step.

- Discarding the length term in the Euler–Lagrange equation, let

$$V(x) = \frac{\partial \phi}{\partial t} = -\lambda_1(u - c_1)^2 + \lambda_2(u - c_2)^2$$

- If $V(x)\phi(x) < 0$ then $\phi(x) = -\phi(x)$.
- There is an anisotropic diffusion step which then handles noise.

This method takes large time steps and so it converges quickly. Finally, there is a decrease in energy at each time step.

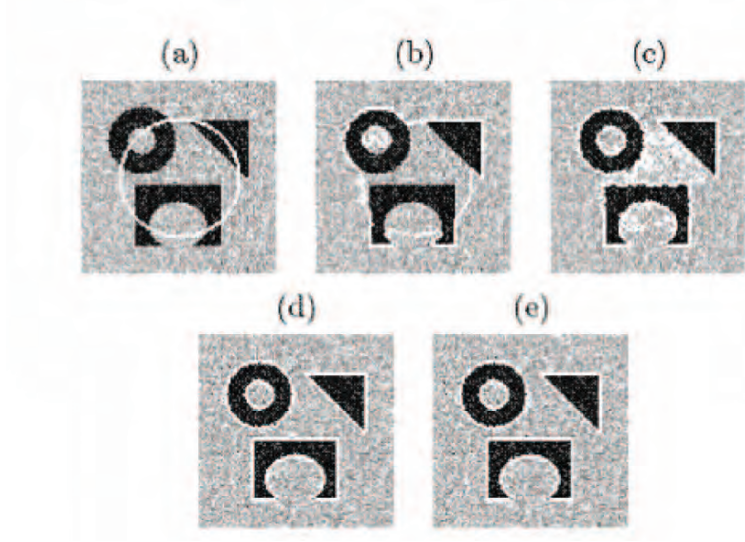


Fig. 27. A synthetic noisy image is segmented in four iterations, which are shown. [55].

5.3 Threshold Dynamics

More recently, work has been done by Esedoglu and Tsai [19], which uses threshold dynamics. This is motivated by a phase-field version of the two-phase piecewise constant Mumford–Shah model. This yields the following gradient-descent equation for u :

$$u_t = 2\epsilon \Delta u - \frac{1}{\epsilon} [W'(u) - 2\lambda [u(c_1 - f)^2 + (u - 1)(c_2 - f)^2]],$$

where $W(\psi) = \psi^2(1 - \psi)^2$.

Using the method developed by Merriman, Bence, and Osher (MBO) [2], the method alternates between a linear parabolic partial differential equation and thresholding:

- Let $v(x) = S(\delta t)u_n(x)$, where $S(\delta t)$ is the propagator of the linear equation

$$w_t = \Delta w - 2\lambda [w(c_1 - f)^2 + (w - 1)(c_2 - f)^2].$$

- Set

$$u_{n+1}(x) = \begin{cases} 0, & \text{if } v(x) \in (-\infty, \frac{1}{2}), \\ 1, & \text{if } v(x) \in (\frac{1}{2}, \infty). \end{cases}$$

This method is fast because the first step is calculated quickly using an fast Fourier transform, and the second step is a threshold. A higher-order scheme has been developed in [18].

6 Acknowledgment

We would like to thank Selim Esedoglu, Richard Tsai, Luminita Vese, Xue-Cheng Tai, and Jason Chung for their support.

References

1. L. Ambrosio and V. Tortorelli. Approximation of functionals depending on jumps by elliptic functionals via γ convergence. *Comp. Applied Math.*, 63:707–711, 1990.
2. B. Merriman, J. K. Bence, and S. J. Osher. Diffusion generated motion by mean curvature. *Proceedings of the Computational Crystal Growers Workshop*, AMS:73–83, 1992.
3. B. Merriman, J. K. Bence, and S. J. Osher. Motion of multiple junctions: A level set approach. *J. Comput. Phys.*, 112, 1994.
4. P. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Comm.*, 31:532–540, 1983.
5. R. E. Caflisch, M. F. Gyure, B. Merriman, S. Osher, C. Rasch, D. D. Vedonsky, and J. J. Zinck. Island dynamics and the level set method for epitaxial growth. *Appl. Math. Letters*, 12(4):13, 1999.
6. S. Casadei, S. Mitter, and P. Perona. Boundary detection in piecewise homogeneous textured images. *Lect. Notes Comput. Sci.*, 588:174–183, 1992.
7. V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Num. Math.*, 66:1–31, 1993.
8. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. J. Comp. Vis.*, 22(1):61–79, 1997.
9. T. Chan, B. Sandberg, and L. Vese. Active contours without edges for vector-valued images. *J. Vis. Comm. Image Represent.*, 11:130–141, 1999.
10. T. Chan, B. Sandberg, and L. Vese. Active contours without edges for textured images. *CAM report 02-39*, 2002.
11. T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 16(2):266–277, 2000.
12. O. Christiansen and X.C. Tai. Fast implementation of the piecewise constant level set method. In *this book*.
13. J. T. Chung and L. A. Vese. Image segmentation using a multilayer level-set approach. *UCLA Cam Report 03-53*, <http://www.math.ucla.edu/applied/cam/index.html>, 2001.
14. M. Clark, A. C. Bovik, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):55–73, 1990.
15. J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filter. *J. Opt. Soc. Am. A*, 2(7):1160–1169, 1985.
16. F. Dibos and G. Koepfler. Color segmentation using a variational formulation. *Actes du 16me Colloque GRETSI*, pages 367–370, 1997.
17. D. Dunn, W. E. Higgins, and J. Wakeley. Texture segmentation using 2-d Gabor elementary functions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(2), 1994.

18. S. Esedoglu, S. Ruuth, and Y. H. Tsai. Threshold dynamics for high order geometric motions. *UCLA CAM Report 06-23*, <http://www.math.ucla.edu/applied/cam/index.html>, 2006.
19. S. Esedoglu and Y. H. Tsai. Threshold dynamics for the piecewise constant Mumford–Shah functional. *J. Comput. Phys.* 211(1):367–384, 2006.
20. T. F.Chan and S. Esedoglu. A multiscale algorithm for Mumford–Shah image segmentation. *UCLA CAM Report 03-77*, 2003.
21. T. F.Chan and S. Esedoglu. Aspects of total variation regularized L^1 function approximation. *SIAM J. Appl. Math.* 65(5):1817–1837, 2005.
22. T. F.Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.*, 66(5):1632–1648, 2006.
23. I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biol. Cybern.*, 61:103–113, 1989.
24. D. Gabor. Theory of communication. *Journal of the Institute of Electrical Engineers*, 93:429–457, 1946.
25. S. Gao and T.D. Bui. A new image segmentation and smoothing model. *ISBI*, pages 137–140, 2004.
26. F. Gibou and R. Fedkiw. A fast hybrid k-means level set algorithm for segmentation. In “4th Annual Hawaii International Conference on Statistics and Mathematics”, pp. 281-291, 2005. Also: *Stanford Technical Report 02-08*, 2002.
27. F. Guichard. A morphological affine and Galilean invariant scale space for movies. *IEEE Trans. Image Process.*, 7(3):444–456, 1998.
28. Portilla J. and Simoncelli E.P. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV*, 40(1):49–71, 2000.
29. A. K. Jain and F. Farrakhonia. Unsupervised texture segmentation using Gabor filters. *Pattern Recogn.*, 23(12):1167–1186, 1991.
30. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contours model. *Int. J. Comput. Vis.*, 1:1167–1186, 1991.
31. S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. *Proc. of ICCV, Cambridge*, pages 810–815, 1995.
32. G. Koepfler, C. Lopez, and J.M. Morel. A multiscale algorithm for image segmentation by variational method. *SIAM J. Numer. Anal.*, 31(1):282–299, 1994.
33. G. Koepfler, C. Lopez, and L. Rudin. Data fusion by segmentation. application to texture discrimination. *Actes du 14me Colloque GRETSI, Juan-les-Pins, September*, pages 707–710, 1993.
34. T.S. Lee, D. Mumford, and A. Yuille. Texture segmentation by minimizing vector-valued energy functionals - the coupled-membrane model. *Lect. Notes Comput. Sci.*, 588:165–173, 1992.
35. J. Lie, M. Lysaker, and X.C. Tai. A binary level set model and some applications to Mumford–Shah image segmentation. *IEEE Trans. Image Process.*, 15(5):1171–1181, 2006.
36. J. Lie, M. Lysaker, and X.C.Tai. A variant of the level set method and applications to image segmentation. *Math. Comp.*, 75:1155–1174, 2006.
37. J. Malik and P. Perona. A computational model of texture segmentation. *Proceedings of CVPR '89*, pages 326–332, 1989.
38. R. Malladi, J. A. Sethian, and B. C. Vemuri. A topology independent shape modeling scheme. *Proc. SPIE Conf. on Geometric Methods in Computer Vision II, San Diego*, 2031:246–258, 1993.

39. G. Dal Maso, J.-M. Morel, and S. Solimini. A variational method in image segmentation. existence and approximation results. *Acta Math.*, 63.
40. M. Moelich and T. Chan. Joint segmentation and registration using logic models. *J. Vis. Commun. Image R.*, 15:333–358, 2005.
41. M. Moelich and T. Chan. Tracking objects with the Chan–Vese algorithm. *CAM Reports 03-14*, [www.math.ucla.edu/applied/cam/index.shtml\(03-14\)](http://www.math.ucla.edu/applied/cam/index.shtml(03-14)), March 2003.
42. M. Moelich. Logic Models for Segmentation and Tracking. *Thesis*, UCLA Mathematics Department, 2004.
43. D. Mumford and J. Shah. Optimal approximation by piecewise-smooth functions and associated variational problems. *Commun. Pure Appl. Math.*, 42:577–685, 1989.
44. S. Osher and R. Fedkiw. Level set methods and dynamic implicit surfaces. *Springer-Verlag*, 2003.
45. S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulation. *J. Comput. Phys.*, 79:12–49, 1988.
46. N. Paragios and R. Deriche. Geodesic active contours for supervised texture segmentation. *Proceedings of the International Conference on Computer Vision and Pattern Recognition, June Fort Collins, Colorado*, 1999.
47. N. Paragios and R. Deriche. Geodesic active regions for supervised texture segmentation. *Proceedings of the 7th International Conference on Computer Vision*, pages 100–115, 1999.
48. C. Sagiv, N.A. Sochen, and Y.Y. Zeevi. Geodesic active contours applied to texture feature space. *M. Kerckhove (Ed.): Scale-Space 2001, LNCS 2106*, pages 344–352, 2001.
49. B. Sandberg and T. Chan. Logic operators for active contours on multi-channel images. *J. Vis. Commun. Image R.*, 16:333–358, 2005.
50. G. Sapiro. Color snakes. *Comput. Vis. Image Understand.*, pages 247–253, 1997.
51. G. Sapiro. Geometric partial differential equations and image analysis. *Cambridge University Press*, 2001.
52. G. Sapiro. Vector (self) snakes: a geometric framework for color, texture, and multiscale image segmentation. *Proc. IEEE ICIP, Lausanne*, 1:817–820, September 1996.
53. G. Sapiro and D. L. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Trans. Image Process.*, 5:1582–1586, 1996.
54. Zhu S.C., Lee T.S., and A.L. Yuille. Region competition: Unifying snakes, region growing, energy/bayes/mdl for multi-band image segmentation. *Proceedings of the IEEE 5th ICCV, Cambridge*, pages 416–423, 1995.
55. B. Song and T. Chan. A fast algorithm for level set based optimization. *UCLA CAM Report 02-68*, <http://www.math.ucla.edu/applied/cam/index.html>, 2002.
56. P. H. Suen and G. Healey. Modeling and classifying color textures using random fields in a random environment. *Pattern Recogn.*, 32(6):1009–1017, 1999.
57. P. H. Suen and G. Healey. The analysis and recognition of real-world textures in three dimensions. *IEEE PAMI*, 22(5):491–503, 2000.
58. X.-C. Tai and C. H. Yao. Image segmentation by piecewise constant Mumford–Shah model without estimating the constants. *J. Comput. Math.*, 24(3):435–443, 2006.

59. A. Tsai, A. Willsky, and A. Yezzi. Curve evolution implementation of the Mumford–Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Trans. Image Process.*, 10(8):1169–1186, August 2001.
60. D.-M. Tsai, S.-K. Wu, and M.-C. Chen. Optimal Gabor filter design for texture segmentation using stochastic optimization. *Image Vis. Comput.*, 19:299–316, 2001.
61. L. Vese and T. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. Comput. Vis.*, 50(3):271–293, 2002.
62. T. P. Weldon and W. E. Higgins. Design of multiple Gabor filters for texture segmentation. *IEEE Int. Conf. Acous. Speech. Sig. Proc.*, vol. IV, pages 2245–2248, 1996.
63. T. P. Weldon and W. E. Higgins. An algorithm for designing multiple Gabor filters for segmenting multi-textured images. *IEEE Conference on Image Processing, Chicago, IL, Oct. 4-7*, 1998.
64. A. Yezzi, L. Zollei, and T. Kapur. A variational approach to joint segmentation and registration. *IEEE Conf. on Comp. Vision and Pattern Recognition*, pages 810–815, 2001.
65. H. K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127:179–195, 1996.
66. S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multi-band image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18:884–900, 1996.

Application of Non-Convex BV Regularization for Image Segmentation

Klaus Frick and Otmar Scherzer

Department of Computer Science, University of Innsbruck, Technikerstr. 21a, 6020 Innsbruck, Austria. E-mail: {klaus.frick, otmar.scherzer}@uibk.ac.at

Summary. In this paper we study a variational image segmentation technique and establish a formal relation to implicit active contour models proposed by Caselles, Catté, Coll and Dibos. The variational method consists in optimization of a non-convex (in the sense of the calculus of variations) functional. We prove well-posedness using convexification and relaxation arguments. We shortly discuss a numerical algorithm for minimization of the variational problem and present some numerical experiments which are compared with segmentations from implicit active contour models.

1 Introduction

In this paper we consider a novel class of *variational level set methods* for *segmentation* of gray valued images. Segmentation denotes the process of determination of interfaces between homogeneous regions. We assume that high variations of the gradient of the image intensity function determine the interfaces. In the recent years starting with the pioneering work of Osher and Sethian [20] *level set methods* have become popular for many areas of applications, such as for instance image segmentation. Typically level set methods for image segmentation are formulated as evolution processes, where the zero level set of the evolving function is used to label the interfaces between the homogeneous regions.

Starting point for this work is a level set method proposed by Caselles, Catté, Coll and Dibos [5], which consists in solving the partial differential equation

$$\partial_t u = g(x)|\nabla u| \left(\nabla \cdot \frac{\nabla u}{|\nabla u|} + \nu \right), \quad (t, x) \in [0, \infty) \times \mathbb{R}^2, \quad (1a)$$

$$u(0, x) = u^0(x), \quad x \in \mathbb{R}^2. \quad (1b)$$

Here ν is a positive parameter and

$$g(x) = \frac{1}{1 + |\nabla K_\sigma * I|^2} \quad (2)$$

where $K_\sigma * I$ denotes the convolution of the original image I with a Gaussian kernel with variance σ^2 . In (1b) a smooth approximation of $1 - \chi_{O'}(x)$ is used for u^0 , where $O' \supseteq O$ is an initial guess of the object O to be segmented. For every point x in the domain of I , the solution $u(t, x)$ of (1) is increasing over time, i.e. regions $\{u(x, t) \leq c\}$ shrink, provided that $(\kappa + \nu)$ stays positive, where $\kappa = \nabla \cdot (\nabla u / |\nabla u|)$ denotes the curvature of u . When approaching the boundary of O , $|\nabla K_\sigma * I|$ becomes large and g is close to zero. There $\partial_t u$ is approximately 0 and the evolution process eventually stagnates. This shows that for $t \rightarrow \infty$ the solution $u(t, x)$ of (1) approximates the interfaces. The stopping function g is close to zero near edges of I , but never vanishes exactly, and therefore (at least theoretically) the level set evolution is never terminating. In order to overcome this drawback Caselles, Kimmel, Sapiro and Sbert [7] proposed the *geodesic active contour* model

$$\partial_t u = |\nabla u| \nabla \cdot \left(g(x) \frac{\nabla u}{|\nabla u|} \right), \quad (t, x) \in [0, \infty) \times \mathbb{R}^2. \quad (3)$$

For given u let us denote by $\mathcal{N}_u = \nabla u / |\nabla u|$, then (3) can be rewritten as

$$\partial_t u = [g(x)\kappa + \nabla g(x)\mathcal{N}_u] |\nabla u|.$$

In comparison with (1) here the term $\nabla g \mathcal{N}_u$ instead of $g\nu$ is used. Near the object borders $-\nabla g$ points towards edges and is close to zero along the boundary. Thus for an evolving curve $u(t, x)$ nearby the boundary $\nabla g \mathcal{N}_u$ is positive and hence forcing the zero level set towards the object (cf. [6]).

Kühne et al. [17] propose the generalized implicit active contour model

$$\partial_t u = g_1(x) |\nabla u| \left(\nabla \cdot \frac{g_2(x) \nabla u}{|\nabla u|} + g_3(x) \right). \quad (4)$$

Both (1) and (3) can be considered special instances of (4). In [17] also efficient numerical algorithms for solving the partial differential equation are provided.

In [23] we analyzed a non-convex variational regularization principle where a minimizer approximates the solution of the *mean curvature motion* equation (MCM). To establish this relation the regularization parameter has to be identified with the evolution time of mean curvature motion. This procedure is analogous as used to compare convex regularization methods and the according evolution processes (cf. [24, 21]). The MCM equation has been studied extensively, for instance in Evans and Spruck [12] and Chen, Giga and Goto [8].

Well-posedness of the non-convex variational regularization principle can be proven using convexification arguments and relaxation from the calculus of variations [14]. The approximation properties of the minimizer have been compared with the solution of the mean curvature motion numerically [14].

However, so far there is a lack of analytical evidence for this relation. Equations such as (4) can be viewed as generalizations of the mean curvature motion equation. In this paper we derive a variational regularization model with asymptotic limit (4). As a byproduct of this analysis, an algorithm for the numerical solution is derived.

The paper is organized as follows: In Section 2 we review existence and uniqueness results for *viscosity solutions* of the evolution equations (1) and (3).

In Section 3 we derive variational regularization models which are formally linked to the evolution processes. Section 4 is concerned with *relaxation methods* for analyzing non-convex regularization functionals. We also derive integral representations for the relaxed functionals. In Section 5 we present some numerical examples and compare the numerical experiments with the results obtained with the PDE methods. We conclude with a brief discussion (cf. Section 6).

The outcome of this paper is that there exist non-convex variational methods, where the minimizers approximate the solutions of implicit active contour models. However, the analysis of both classes of segmentation algorithms is completely different: while the evolution processes are analyzed within the framework of viscosity solutions, the non-convex regularization methods are analyzed in a functional analytical framework of the calculus of variations. In particular the latter allows to characterize the minimizers as functions of bounded total variation.

2 Review on the Mathematical Analysis of Evolution Processes

Existence and uniqueness of solutions of evolution equations such as (4) are usually analyzed in the framework of *viscosity solutions*. For more background on this topic we refer to [9].

In [5, Thm. 3.1] existence and uniqueness of a viscosity solution of (1) has been proven. We recall the result: Let $\mathcal{C}_0(X)$ denote this space of bounded uniformly continuous functions on a set X .

Theorem 1. [5, Thm. 3.1] *Let $u^0, v^0 \in \mathcal{C}_0(\mathbb{R}^2) \cap W^{1,\infty}(\mathbb{R}^2)$ and $g, g^{1/2} \in W^{1,\infty}(\mathbb{R}^2)$ and $g \geq 0$. Then Equation (1) has a unique viscosity solution*

$$u \in \mathcal{C}([0, \infty) \times \mathbb{R}^2) \cap L^\infty([0, T], W^{1,\infty}(\mathbb{R}^2)), \quad T < \infty,$$

satisfying

$$\inf_{x \in \mathbb{R}^2} u^0(x) \leq u(t, x) \leq \sup_{x \in \mathbb{R}^2} u^0(x). \quad (5)$$

Moreover, if v is the viscosity solution of (1) with initial data v^0 , then for all $T \in [0, \infty)$ we have

$$\sup_{0 \leq t \leq T} \|u(x, t) - v(x, t)\|_{L^\infty} \leq \|u^0(x) - v^0(x)\|_{L^\infty}.$$

Existence and uniqueness of a viscosity solution of (3) has been proven in [6]:

Theorem 2. [6, Thm. 3] *Let $u^0, v^0 \in C_0(\mathbb{R}^2) \cap W^{1,\infty}(\mathbb{R}^2)$ and $g \in W^{2,\infty}(\mathbb{R}^2)$, $g^{1/2} \in W^{1,\infty}(\mathbb{R}^2)$ and $g \geq 0$. Then Equation (3) has a unique viscosity solution*

$$u \in \mathcal{C}([0, \infty) \times \mathbb{R}^2) \cap L^\infty([0, T], W^{1,\infty}(\mathbb{R}^2)), \quad \forall T < \infty,$$

satisfying

$$\inf_{x \in \mathbb{R}^2} u^0(x) \leq u(t, x) \leq \sup_{x \in \mathbb{R}^2} u^0(x). \quad (6)$$

Moreover, if v is the viscosity solution of (1) with initial data v^0 , then for all $T \in [0, \infty)$ we have

$$\sup_{0 \leq t \leq T} \|u(x, t) - v(x, t)\|_{L^\infty} \leq \|u^0(x) - v^0(x)\|_{L^\infty}.$$

For an alternative proof see [16, Thm.19.2]. An analogous result to Theorem 2 also holds for the equation

$$\partial_t u = g(x)(\kappa + \nu)|\nabla u| + \nabla g(x) \nabla u, \quad \nu > 0. \quad (7)$$

Moreover [6, Thm.5] shows geometrical correctness of model (7):

Theorem 3. [6, Thm. 5] *Let g be as in Theorem 2 and assume that $\Gamma = \{x \in \mathbb{R}^2 : g(x) = 0\}$ is a simple Jordan curve of class C^2 such that $\nabla g = 0$ along Γ . Moreover let $u^0 \in C^2(\mathbb{R}^2)$ such that $\Gamma \cup \text{int}(\Gamma) \subseteq \{x \in \mathbb{R}^2 : u^0(x) \leq 0\}$. Let $u(t, x)$ denote the unique solution of (7) and set $\Gamma(t) = \{x \in \mathbb{R}^2 : u(t, x) = 0\}$. Then for ν sufficiently large*

$$\Gamma(t) \rightarrow \Gamma, \quad \text{as } t \rightarrow \infty$$

with respect to the Hausdorff distance.

Proof. [7, Thm. 3].

Moreover, in [7] the corresponding results for space dimension three are available.

3 Variational Level Set Model for Image Segmentation

In this section we establish a formal relation between the curvature driven evolution equation (4) and iterative *level set regularization* techniques. This derivation essentially follows [14, 23].

Let $\mathcal{S}$ be a space of real valued functions defined on a bounded domain $\Omega \subseteq \mathbb{R}^n$ with Lipschitzian boundary $\partial\Omega$.

For given functions $g_1, g_2, g_3 \in L^\infty(\Omega)$, $g_1 > 0$ and $u^0 \in L^\infty(\Omega)$ and fixed regularization parameter $\alpha > 0$ we consider the functional $I : \mathcal{S} \rightarrow \mathbb{R} \cup \{+\infty\}$ defined by

$$I(u; \alpha, u^0) := \int_{\Omega} \frac{S(u, u^0)}{g_1} + \alpha (g_2 |\nabla u| - g_3 u) dx, \quad (8)$$

where

$$S(u, v) = \frac{(u - v)^2}{2|\nabla u|}.$$

and assume that a minimizer $v \in \mathcal{S}$ can be characterized by the formal optimality condition

$$u^0 \in u + \alpha A(u), \quad (9)$$

where

$$\begin{aligned} A(u; \alpha, u^0) &= g_1 |\nabla u| \left(\nabla \cdot \left[\left(\frac{(u - u^0)^2}{2|\nabla u|^2 g_1 \alpha} - g_2 \right) \frac{\nabla u}{|\nabla u|} \right] + g_3 \right) \\ &= g_1 |\nabla u| \left(\nabla \cdot \left[(T(u; \alpha, u^0) - g_2) \frac{\nabla u}{|\nabla u|} \right] + g_3 \right). \end{aligned} \quad (10)$$

To see the formal relation to the evolution process (4) let $t > 0$, $n \in \mathbb{N}$, $\Delta t = t/n$ and $u_0^n := u^0$. We proceed iteratively and denote by $u_k^n \in \mathcal{S}$, $k = 1, \dots, n$, a minimizer of $I(u; \Delta t, u_{k-1}^n)$ (presumably it exists). We define a time dependent function $\mathbf{u}(t, x)$

$$\mathbf{u}(t, \cdot) = \lim_{n \rightarrow \infty} u_n^n(\cdot).$$

Then, from (9) it follows that

$$\frac{\mathbf{u}(t, x) - \mathbf{u}(t - \Delta t, x)}{\Delta t} \in -A(\mathbf{u}(t, x); \Delta t, \mathbf{u}(t - \Delta t, x)). \quad (11)$$

Taking the limit $\Delta t \rightarrow 0^+$ in (11) and noting that at least in a formal sense

$$\begin{aligned} \lim_{\Delta t \rightarrow 0^+} T(\mathbf{u}(t, x); \Delta t, \mathbf{u}(t - \Delta t, x)) &= \lim_{\Delta t \rightarrow 0^+} \left(\frac{(\mathbf{u}(t, x) - \mathbf{u}(t - \Delta t, x))^2}{2|\nabla \mathbf{u}(t, x)|^2 g_1} \right) \Delta t \\ &= \lim_{\Delta t \rightarrow 0^+} \left(\frac{\partial_t \mathbf{u}(t, x)^2}{2|\nabla \mathbf{u}(t, x)|^2 g_1} \right) \Delta t = 0 \end{aligned}$$

we get

$$\partial_t u = g_1 |\nabla u| \left(\nabla \cdot \left[g_2 \frac{\nabla u}{|\nabla u|} \right] + g_3 \right), \quad (12a)$$

$$u(0, x) = u^0(x). \quad (12b)$$

This derivation shows a formal relation between (12a) and the regularization method of minimizing (8). For g as in (2) the settings $(g_1 = g, g_2 \equiv 1, g_3 \equiv \nu)$ and $(g_1 \equiv 1, g_2 = g, g_3 \equiv 0)$ relate (12a) to (1) and (3) respectively.

In the following sections we prove well-posedness of the regularization functional (8), i.e., the existence of a minimizer, and a practical approach for minimization of this functional.

4 Relaxation

We prove well-posedness of the functional $I(u; \alpha, u^0)$ on the space $BV(\Omega)$, the space of functions of bounded variation. Properties and an extensive discussion of the space $BV(\Omega)$ can be found in Evans and Gariepy [11]. In general, for $u \in BV(\Omega)$, Du (the derivative of u) is a signed, $\mathbb{R}^n$ -valued, Radon measure. The evaluation of a convex function of a signed Radon measure is well-defined (cf. Temam [25]). In our situation, the function $A \rightarrow \frac{(u - u^0(x))^2}{2|A|}$ is not convex and therefore in general the functional $S(u, u^0)$ is not defined. To cope with this problem we can proceed with the Radon-Nikodym theorem and decompose the measure Du into its absolutely continuous and singular parts, i.e. $Du = \nabla u dx + D^s u$. If the singular part does not vanish, we can expect high gradients and thus $S(u, u^0)$ can be suspected to be small. This suggests the following interpretation of the functional $I(u)$ defined in (8):

$$I(u) := \int_{\Omega} \frac{S(u, u^0)(x)}{g_1(x)} - \alpha g_3(x) u(x) dx + \alpha \int_{\Omega} g_2(x) d|Du|(x). \quad (13)$$

where we use the convention that

$$S(u, u^0) = 0, \quad \text{if } u = u^0.$$

Note that in (13) the functional $\int_{\Omega} \frac{S(u, u^0)(x)}{g_1(x)} - \alpha g_3(x) u(x) dx$ is defined by avoiding the singular parts of Du . Only the functional $\int_{\Omega} g_2(x) d|Du|(x)$ takes into account the singular parts.

In the classical theory of the *calculus of variations* (cf. Dacorogna [10]) existence of minimizers of a functional $\int_{\Omega} f(x, u(x), \nabla u(x)) dx$ in a Banach space is derived from *lower semi continuity* and *coercivity* of the functional. The lower semi continuity is closely related to convexity of the functional f with respect to the variable A , which is not available for our example. And in fact it turns out that the functional $I(u)$ is not lower semi continuous with respect to the L^1 topology on $BV(\Omega)$.

For showing well-posedness of minimizers of *non-convex* energy functionals, a common approach is by *relaxation* (cf. Buttazzo [4]).

Definition 1. Let $\emptyset \neq X \subseteq BV(\Omega)$ and $I : X \rightarrow \mathbb{R} \cup \{+\infty\}$. The relaxation $\mathcal{R}(I, X) : BV(\Omega) \rightarrow \mathbb{R} \cup \{+\infty\}$ of I is defined by

$$\mathcal{R}(I, X)(u) = \begin{cases} +\infty & \text{if } u \notin \overline{X} \cap BV(\Omega), \\ \inf \{ \liminf_{k \rightarrow \infty} I(u_k) : \{u_k\}_{k \in \mathbb{N}} \subseteq X, \|u_k - u\|_{L^1} \rightarrow 0 \}. \end{cases} \quad (14)$$

The closure of X is taken with respect to the L^1 -topology. Moreover, we use the abbreviation $\mathcal{R}(I) := \mathcal{R}(I, BV(\Omega))$.

Lemma 1. Let $I : BV(\Omega) \rightarrow \mathbb{R} \cup \{+\infty\}$ and $\tilde{I} : BV(\Omega) \rightarrow \mathbb{R}$.

1. $\mathcal{R}(I)$ is lower semi continuous on $BV(\Omega)$ w.r.t. L^1 .
2. $\mathcal{R}(\tilde{I})$ is proper, i.e. $\mathcal{R}(\tilde{I}) \not\equiv \infty$.
3. If $\tilde{I}$ is continuous w.r.t. L^1 we have $\mathcal{R}(I + \tilde{I}) = \mathcal{R}(I) + \tilde{I}$.

The proof follows immediately from the definition of the relaxation.

Relaxation is an important tool in the analysis of non-convex energy functionals. However, for numerical minimization of $I(u)$ it is not appropriate, since it requires to calculate for each function $u \in BV(\Omega)$ the relaxed functional value by approximating u by *all* sequences converging to u , before it can be optimized. In the following we derive an integral representation for the relaxation of I , which then can be solved efficiently numerically. The integral representation is convex with respect to the gradient variable and this allows to prove that the functional is lower semi continuous. Under certain assumptions the functional is also coercive and thus it attains a minimizer. Moreover, the minimizer of this functional can be consider a *generalized* minimizer of the relaxed functional. According to Lemma 1(3), $\mathcal{R}(I) = \mathcal{R}(J) - \alpha \int_{\Omega} g_3(x)u(x)dx$. Thus it suffices to calculate the relaxation $\mathcal{R}(J)$ of the functional

$$J(u) = \int_{\Omega} \frac{S(u, u^0)(x)}{g_1(x)} dx + \alpha \int_{\Omega} g_2(x) d|Du|(x). \quad (15)$$

In the following we derive the integral representation of the relaxation of the functional J restricted on a subspace of Sobolev functions. Here, we use some results from [15].

Lemma 2. Assume that there exists constants $C_1, C_2 > 0$ such that

$$\frac{1}{C_i} \leq g_i(x) \leq C_i, \quad i = 1, 2, \forall x \in \Omega, \quad (16)$$

and that g_2 is continuous. Then, for $u \in W^{1,1}(\Omega)$ we have

$$\mathcal{R}(J, W^{1,1}(\Omega))(u) = J^c(u),$$

where

$$J^c(u) = \int_{\Omega} f^c(x, u(x), \nabla u(x)) dx, \quad (17a)$$

$$f^c(x, \xi, A) = \begin{cases} \frac{(\xi - u^0)^2}{2|A|g_1} + \alpha g_2|A| & \text{if } \sqrt{2\alpha g_1 g_2}|A| > |\xi - u^0|, \\ \sqrt{\frac{2\alpha g_2}{g_1}}|\xi - u^0| & \text{else.} \end{cases} \quad (17b)$$

Proof. We apply the results in [15] and therefore we adopt the notation there. To this end we write for $u \in W^{1,1}(\Omega)$

$$J(u) = \int_{\Omega} f(x, G(u, u^0)(x), \nabla u(x)) dx,$$

with

$$f(x, \xi, A) = \frac{\xi^2}{|A|} + \alpha g_2(x)|A| \text{ and } G(u, v)(x) = \frac{|u(x) - v(x)|}{\sqrt{2g_1(x)}}.$$

Note that under our assumptions the operator

$$G : W^{1,1}(\Omega) \times L^1(\Omega) \rightarrow L^1(\Omega)$$

is continuous with respect to the L^1 -norm.

Since g_2 is continuous, for every $x \in \Omega$ there exists $r > 0$ and a continuous function $\eta : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, satisfying

$$\eta(0) = 0 \text{ and } |g_2(x) - g_2(x')| \leq \eta(|x - x'|) \text{ for all } x' \in B_r(x).$$

and thus

$$|g_2(x)|A| - g_2(x')|A|| \leq \eta(|x - x'|)(1 + g_2(y)|A|)$$

holds for all $x', y \in B_r(x)$. From these inequalities the conditions (S1) and (S2) in [15] follow. Moreover, since $f(x, \xi, A) \geq \alpha g_2(x)|A|$ and $f(x, \xi, A) \geq 2\sqrt{\alpha g_2(x)|\xi|}$ we have

$$f(x, \xi, A) \geq \frac{1}{2} \min \left\{ 2\sqrt{\frac{\alpha}{C_2}}, \frac{\alpha}{C_2} \right\} (|\xi| + |A|).$$

Thus we can apply the first part of [15, Thm. 2.3] and obtain the assertion.

Remark 1. The continuity assumption of g_2 can be relaxed. It suffices to find a positive continuous mapping $\gamma : \Omega \rightarrow \mathbb{R}$ which is bounded from below and above by positive constants (cf. (16)) such that

$$|f(x, \xi, A) - f(x', \xi', A')| \leq \left| \frac{\xi^2}{|A|} + \alpha \gamma(x)|A| - \frac{\xi'^2}{|A'|} - \alpha \gamma(x')|A'| \right|,$$

and

$$a\gamma(x) \leq g_2(x) \leq \gamma(x),$$

for all $(x, \xi, A), (x', \xi', A') \in \Omega \times \mathbb{R} \times \mathbb{R}^n \setminus \{0\}$ and a constant $a > 0$ (cf. [15]).

In the following we derive an integral representation of the functional

$$J^*(u) = \begin{cases} J(u) & \text{if } u \in W^{1,1}(\Omega), \\ +\infty & \text{else.} \end{cases} \quad (18)$$

We mention that $J \neq J^*$ but $\mathcal{R}(J^*) = \mathcal{R}(J)$ (cf. Theorem 4 below).

Taking into account Lemma 2 the proof of the following Lemma is along the lines of the proof of [14, Thm. 2].

Lemma 3. For $u \in W^{1,1}(\Omega)$ and $r > 0$ let

$$X^r = \{u \in BV(\Omega) : \|u\|_\infty < r\}.$$

and

$$J^{c,r}(u) = \int_{\Omega} f^{c,r}(x, u(x), \nabla u(x)) dx,$$

where

$$f^{c,r}(x, \xi, A) = \begin{cases} \frac{(\xi - u^0)^2 \wedge r^2}{2|A|g_1} + \alpha g_2 |A| & \text{if } \sqrt{2\alpha g_1 g_2} |A| > |\xi - u^0| \wedge r, \\ \sqrt{\frac{2\alpha g_2}{g_1}} (|\xi - u^0| \wedge r) & \text{else.} \end{cases},$$

where $a \wedge b = \max(a, b)$. With $r^0 = \|u^0\|_\infty$ it follows that

$$\mathcal{R}(J^*)(u) = \mathcal{R}(J^{c,r}, W^{1,1}(\Omega))(u), \quad u \in X^{r-r^0}.$$

In the following we require a generalization of the functional $J^c(u)$ for functions $u \in BV(\Omega)$. For this, we use the Lebesgue decomposition of the signed measure $Du = \nabla u dx + D^s u$ and define

$$J^c(u) = \int_{\Omega} f^c(x, u(x), \nabla u(x)) dx. \quad (19)$$

Note that for $u \in BV(\Omega)$, J^c does not depend on the singular parts of the measure Du .

For functionals $\Phi : BV(\Omega) \rightarrow \mathbb{R}$ integral representations of $\mathcal{R}(\Phi, W^{1,1}(\Omega))$ have been studied by Bouchitté, Fonseca and Mascarenhas [2].

Lemma 4. Let the assumptions of Lemma 2 hold. Then for all $u \in X^{r-r^0}$,

$$\mathcal{R}(J^{c,r}, W^{1,1}(\Omega))(u) = J^c(u) + \alpha \int_{\Omega} g_2(x) d|D^s u|(x).$$

Recall that $J^c(u)$ has to be understood in the sense of (19).

Proof. The result of this lemma is a consequence of Theorem 4.1.4. in Bouchitté, Fonseca and Mascarenhas [2]. To apply this result general assumptions for $f^{c,r}(x, \xi, A)$ have to be verified (cf. [2, Section 4.1]):

1. From the definition of $f^{c,r}$ it follows that

$$\frac{\alpha}{C_2} |A| \leq f^{c,r}(x, \xi, A) \leq \sqrt{\frac{\alpha C_1 C_2}{2}} r + \alpha C_2 |A|. \quad (20)$$

2. In [13, Remark 5.1] it has been shown that

$$|f^{c,r}(x, \xi, A) - f^{c,r}(x, \eta, A)| \leq 3C_1 \sqrt{\frac{\alpha C_1 C_2}{2}} \delta, \text{ for } |\xi - \eta| < \delta. \quad (21)$$

3. For $|A| = 1$, the *recession function*

$$f_{\infty}^{c,r}(x, \xi, A) := \limsup_{s \rightarrow \infty} f^{c,r}(x, \xi, sA)/s = \alpha g_2(x)A$$

satisfies

$$\left| f_{\infty}^{c,r}(x, \xi, A) - \frac{f^{c,r}(x, \xi, sA)}{s} \right| = \frac{(\xi - u^0)^2 \wedge r^2}{2s^2 g_1} \leq \frac{r^2 C_1}{2} s^{-2}, \quad (22)$$

If s is large enough we observe that

$$\left| f_{\infty}^{c,r}(x, \xi, A) - \frac{f^{c,r}(x, \xi, sA)}{s} \right| \leq \frac{C}{s^m},$$

for a constant C and $0 < m < 1$, i.e. condition (H4) in [2] is satisfied.

These three estimates and the continuity of g_2 allow us to apply [2, Thm. 4.1.4], which shows that

$$\begin{aligned} \mathcal{R}(J^{c,r}, W^{1,1}(\Omega))(u) &= J^c(u) + \alpha \int_{\Omega \cap S_u} g_2(x) [u](x) d\mathcal{H}^{n-1}(x) \\ &\quad + \alpha \int_{\Omega} g_2(x) d|C(u)|(x) = I^c(u) + \alpha \int_{\Omega} g_2(x) d|D^s(u)|(x). \end{aligned}$$

Here $[u](x) = u^+(x) - u^-(x)$, where $u^+(x)$ and $u^-(x)$ are the approximate upper and lower limits of u and $S_u = \{x \in \Omega : u^+(x) > u^-(x)\}$ denotes the jump set. $D^s u$ and $C(u)$ denote the singular part and Cantor part of the measure Du (cf. [11, Sec. 5.9]) respectively.

To apply Theorem 4.1.4. in Bouchitté, Fonseca and Mascarenhas [2] the assumption $u \in X^{r-r^0}$ is essential and used to prove the estimates (20) and (22).

In the following we derive the integral representation of the functional I for all functions in $BV(\Omega)$ by considering the limiting case of the functionals $J^{c,r}$ on X^{r-r^0} when $r \rightarrow \infty$.

Lemma 5. *Assume that $u \in BV(\Omega)$ and $g_2 \in \mathcal{C}_c(\Omega)$. Then*

$$\mathcal{R}(J^*)(u) = J^c(u) + \alpha \int_{\Omega} g_2(x) d|D^s u|(x).$$

Proof. Since

$$f^c(x, \xi, 0) \leq \sqrt{2\alpha C_1 C_2} (\|u^0\|_{\infty} + |\xi|) \text{ for all } (x, \xi) \in \Omega \times \mathbb{R},$$

we can apply Proposition 2.4. of Buttazzo and DalMaso [3] and get

$$\mathcal{R}(J^*)(u) = \lim_{r \rightarrow \infty} \mathcal{R}(J^*)(u^{(r)}) = \lim_{r \rightarrow \infty} \left[J^c(u^{(r)}) + \alpha \int_{\Omega} g_2(x) d|D^s u^{(r)}|(x) \right].$$

Since $f^c(x, u^{(r)}(x), \nabla u^{(r)}(x))$ is increasing with r for all $x \in \Omega$ it follows from the monotone convergence theorem that

$$J^c(u^{(r)}) \rightarrow J^c(u) \text{ for } r \rightarrow \infty.$$

To prove the assertion, we have to show that

$$\lim_{r \rightarrow \infty} \int_{\Omega} g_2(x) d|Du^{(r)}|(x) = \int_{\Omega} g_2(x) d|Du|(x). \quad (23)$$

Let $U \subseteq \Omega$ be open. Since the functional $u \rightarrow |Du|(U)$ defined on $BV(\Omega)$ is lower semi continuous w.r.t. to the $L^1(\Omega)$ -norm it follows that

$$\liminf_{r \rightarrow \infty} |Du^{(r)}|(U) \geq |Du|(U).$$

Moreover from the coarea formula on $BV(\Omega)$ (see e.g. [11, Thm.1, Chap. 5.5.]) it follows that

$$\limsup_{r \rightarrow \infty} |Du^{(r)}|(U) \leq |Du|(U).$$

Hence we have

$$|Du^{(r)}|(U) \rightarrow |Du|(U). \quad (24)$$

Set $U_t = \{x \in \Omega : g_2(x) > t\}$ and $\mu^r(t) = |Du^{(r)}|(U_t)$. Since g_2 is continuous, U_t is open and together with [22, Thm. 8.16], (24), and the monotone convergence theorem we find that

$$\lim_{r \rightarrow \infty} \int_{\Omega} g d|Du^{(r)}| = \lim_{r \rightarrow \infty} \int_0^{\infty} \mu^r(t) dt = \int_0^{\infty} \lim_{r \rightarrow \infty} \mu^r(t) dt = \int_{\Omega} g d|Du|.$$

Setting $\Omega_r = \{x \in \Omega : u^{(r)}(x) = u(x)\}$ it follows from [1, Expl. 3.100] that

$$|Du^{(r)}|(\Omega \setminus S_u) = |Du|(\Omega_r \setminus S_u),$$

from which it follows again from the monotone convergence that

$$\lim_{r \rightarrow \infty} \int_{\Omega} g_2 |\nabla u^{(r)}| dx = \int_{\Omega} g_2 |\nabla u| dx, \quad (25)$$

where $|\nabla u^{(r)}|$ denoted the density of the absolutely continuous part of the $|Du^{(r)}|$. From (4), (25) and the fact that $|D^s u^{(r)}| = |Du^{(r)}| - |\nabla u^{(r)}| dx$, (23) (and thus the assertion of this lemma) follows.

Using the previous result we can state the integral representation of the functional I on $BV(\Omega)$.

Theorem 4. Assume that $u \in BV(\Omega)$, $u^0 \in L^\infty(\Omega)$, and $g_2 \in \mathcal{C}(\Omega)$. Moreover assume that $g_1, g_2 \in L^\infty(\Omega)$ satisfy (16). Then

$$\mathcal{R}(I)(u) = J^c(u) + \alpha \int_{\Omega} g_2(x) d|D^s u|(x) - \alpha \int_{\Omega} g_3(x) u dx. \quad (26)$$

Proof. From Lemma 1 it follows that $\mathcal{R}(I) = \mathcal{R}(J) - \alpha \int_{\Omega} g_3(x) u dx$. Therefore, in order to find the integral representation of $\mathcal{R}(I)$ it suffices to calculate the integral representation of $\mathcal{R}(J)$. From the definition of J^* (cf. (18)) and J^c (cf. (17)) and Lemma 4 it is evident that

$$J^c(u) \leq J(u) \leq J^*(u) \text{ for all } u \in \text{BV}(\Omega) .$$

Therefore also

$$\mathcal{R}(J^c)(u) \leq \mathcal{R}(J)(u) \leq \mathcal{R}(J^*)(u) \text{ for all } u \in \text{BV}(\Omega) .$$

From Lemma 5 we have $\mathcal{R}(J^*)(u) = J^c(u)$ and thus $\mathcal{R}(J^c)(u) = \mathcal{R}(J^*)(u) = J^c(u)$. This shows the assertion.

Lemma 6. Assume that g_1, g_2 satisfy the assumptions of Theorem 4 and that

$$|g_3(x)| < \sqrt{\frac{g_2(x)}{2\alpha g_1(x)}}, \quad x \in \Omega. \quad (27)$$

Then $\mathcal{R}(I)$ is coercive on $BV(\Omega)$.

Proof. Let $(x, \xi, A) \in \Omega \times \mathbb{R}^+ \times \mathbb{R}^n$. Below, we show that

$$f^c(x, \xi, A) - \alpha g_3 \xi \geq \gamma(|A| + |\xi|) - \varepsilon |u^0(x)|, \quad (28)$$

with appropriate constants $\gamma := \gamma(g_1, g_2, g_3, \alpha)$ and $\varepsilon := \varepsilon(g_1, g_2, g_3, \alpha)$. Taking the minimum of f^c with respect A shows that

$$\begin{aligned} f^c(x, \xi, A) - \alpha g_3 \xi &= \frac{(\xi - u^0(x))^2}{2|A|g_1} + \alpha g_2 |A| - \alpha g_3 \xi \\ &\geq \sqrt{\frac{2\alpha g_2}{g_1}} |\xi - u^0(x)| - \alpha g_3 \xi . \end{aligned} \quad (29)$$

We differ between two cases:

1. If $\sqrt{2\alpha g_1 g_2} |A| > |\xi - u^0(x)|$, then from (29) it follows that

$$\sqrt{\frac{2\alpha g_2}{g_1}} (|\xi - u^0(x)|) - \alpha g_3 \xi \geq \underbrace{\left[\sqrt{\frac{2\alpha g_2}{g_1}} - \alpha \text{sgn}(g_3 \xi) |g_3| \right]}_{>0} |\xi| - \sqrt{\frac{\alpha g_2}{2g_1}} |u^0(x)| .$$

Moreover,

$$\begin{aligned} f^c(x, \xi, A) - \alpha g_3 \xi &> \alpha g_2 |A| - \alpha g_3 \xi \\ &\geq \alpha g_2 |A| - \alpha \text{sgn}(g_3 \xi) |g_3| |\xi - u^0(x)| - \alpha |g_3| |u^0(x)| \\ &\geq \alpha \underbrace{\left[g_2 - \text{sgn}(g_3 \xi) |g_3| \sqrt{2\alpha g_1 g_2} \right]}_{>0} |A| - \alpha |g_3| |u^0(x)| . \end{aligned}$$

Summing up the two estimates shows (28).

2. If $\sqrt{2\alpha g_1 g_2}|A| \leq |\xi - u^0(x)|$ we find that

$$\begin{aligned} f^c(x, \xi, A) - \alpha g_3 \xi &= \sqrt{\frac{2\alpha g_2}{g_1}} |\xi - u^0(x)| - \alpha g_3 \xi \\ &\geq \underbrace{\left[\sqrt{\frac{2\alpha g_2}{g_1}} - \alpha \operatorname{sgn}(\xi) |g_3| \right]}_{=\beta > 0} |\xi| - \sqrt{\frac{2\alpha g_2}{g_1}} |u^0(x)| \\ &\geq \beta \sqrt{2\alpha g_1 g_2} |A| - \left(\sqrt{\frac{2\alpha g_2}{g_1}} + \beta \right) |u^0(x)|. \end{aligned}$$

This again shows (28).

Using these pointwise estimates it follows together with Theorem 4 that

$$\begin{aligned} \mathcal{R}(I)(u) &= \int_{\Omega} f^c(x, u(x), \nabla u(x)) - \alpha g_3(x) u(x) dx + \alpha \int_{\Omega} g_2(x) d|D^s u|(x) \geq \\ &\gamma(\|u\|_{L^1} + \|\nabla u\|_{L^1}) - \varepsilon \|u^0\|_{L^1} + \frac{\alpha}{C_2} |D^s u|(\Omega) \geq \gamma' \|u\|_{BV} - \varepsilon \|u^0\|_{L^1}, \end{aligned}$$

with some appropriate positive constants γ', γ and ε .

The following theorem guarantees well-posedness of the relaxed functional.

Theorem 5. *Assume that g_1, g_2, g_3 satisfy the assumptions of Lemma 6 and that $u^0 \in L^\infty(\Omega)$. Then there exists $u \in BV(\Omega)$ such that*

$$\mathcal{R}(I)(u) = \inf\{\mathcal{R}(I)(v) : v \in BV(\Omega)\}. \quad (30)$$

Proof. Let $\lambda = \inf\{\mathcal{R}(I)(v) : v \in BV(\Omega)\}$ and $\{u_k\}_{k \in \mathbb{N}} \subseteq BV(\Omega)$ such that $\lim_{k \rightarrow \infty} \mathcal{R}(I)(u_k) = \lambda$. Since $\mathcal{R}(I)$ is coercive (cf. Lemma 6), the sequence $\{u_k\}_{k \in \mathbb{N}}$ is uniformly bounded in $BV(\Omega)$. Hence there exists a subsequence $\{u_{k'}\}_{k' \in \mathbb{N}}$ and $u \in BV(\Omega)$ such that $\|u_{k'} - u\|_{L^1} \rightarrow 0$ (cf. [11, Sec. 5.1, Thm. 4]). From the lower semi continuity of $\mathcal{R}(I)$ (cf. Lemma 1) it follows that $\lambda \geq \mathcal{R}(I)(u)$. Since $\lambda \leq \mathcal{R}(I)(v)$ for all $v \in BV(\Omega)$ the assertion follows.

The above theorem states that minimization of $\mathcal{R}(I)$ is well-posed.

From [14, Thm. 2] we know that if a minimizer of I exists it is also a minimizer of $\mathcal{R}(I)$. If the minimizer of I does not exist, then there exist at least a sequence of functions $\{v_n\}_{n \in \mathbb{N}}$ in $BV(\Omega)$ such that $I(v_n) \rightarrow \mathcal{R}(I)(v)$, where $v = \operatorname{argmin} \mathcal{R}(I)$. This shows that the minimizer of $\mathcal{R}(I)$ can be considered as a generalized minimizer of I .

5 Numerical Simulations

In this section we present some numerical experiments comparing the solutions of the geometric equations (1) and (3) with the variational technique of minimization of $\mathcal{R}(I)$.

In order to compute a minimizer of $\mathcal{R}(I)$ in (13) we use a gradient descent method and solve the weak optimality condition

$$\langle \partial_\tau \tilde{u}(\tau) - \partial \mathcal{R}(I)(\tilde{u}(\tau)), v \rangle_{L^2} = 0, \quad v \in W_c^{1,2}(\Omega) \quad (31)$$

up to a stationary point and set $u = \lim_{\tau \rightarrow \infty} \tilde{u}(\tau)$. We use a *semi implicit* finite element approach for the solution of (31), i.e. in the n -th iteration step (with respect to the artificial time evolution τ) non-linear terms depending on u are evaluated at $u^{(n-1)}$. The resulting linear system is solved by a CG method. A detailed description of the numerical implementation is intended to be discussed in a forthcoming paper.

For the solution of the evolution equation (12) we use a semi implicit *additive operator splitting (AOS)* technique (cf. Lü et al. [18, 19] and Weickert [26]). A semi implicit numerical scheme for solving (12) can be implemented as follows:

$$u^{n+1} = \left(I - \Delta t \sum_{i=1}^d A_i(u^n) \right)^{-1} u^n, \quad (32)$$

where the operators A_i are discrete approximations of the space derivatives in (12). Here d denotes the space dimension. The AOS technique is a first order approximation of (32) which reads as follows:

$$u^{n+1} = \frac{1}{n} \sum_{i=1}^d (I - n \Delta t A_i(u^n))^{-1} u^n. \quad (33)$$

Unlike (32), Equation (33) has the advantage that solving for u^{n+1} reduces to solving n tridiagonal systems, which can be implemented very efficiently (cf. [27]). Weickert and Kühne [17] have introduced a fast AOS scheme for solving (12) and presented a series of numerical experiments.

In the following examples we consider the initial contours as boundaries of sets C , that enclose the object to recover. Rather than the characteristic function, as proposed in Caselles, Catté, Coll and Dibos [5], we use the *signed distance function*

$$x \rightarrow \begin{cases} \text{dist}(x, C) & \text{if } x \in \text{ext}(C) \\ 0 & \text{if } x \in \partial C \\ -\text{dist}(x, C) & \text{else} . \end{cases}$$

as initial value u^0 for the partial differential equations (1), (3), and (13).

In order to compare the results obtained with implicit active contour models with the results of the variational techniques proposed in this paper, we first study the simple example shown in Figure 1. The original image g_0 is the characteristic function of a star shaped set. We choose u^0 the signed distance function of a disc outside the star shaped domain. Figure 2 (top) shows the numerical solutions of (1) with



Fig. 1. Original image g_0 (star shaped) and initial contour.

$$g := \frac{1}{\varepsilon + |\nabla g_0|}, \quad \nu = 1/4 \text{ and } \varepsilon = 1. \quad (34)$$

at $t = 40, 80, 120$ and 200 . The zero isolines of the corresponding solutions $u(t)$ are plotted.

In the regularization technique for minimization of $\mathcal{R}(I)$ we selected $g_1 = g$, $g_2 \equiv 1$, and $g_3 \equiv \nu$. Therefore the coercivity condition (27) becomes $\alpha \leq 8$. The bottom line in Figure 2 shows the (iterated) minimizers of (13) for $\alpha = 8$

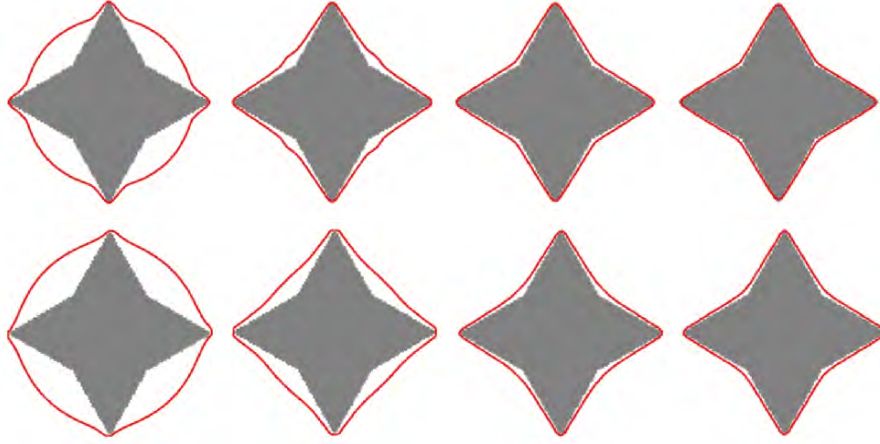


Fig. 2. Top: Solutions of (1) for $t = 40, 80, 120$ and 200 . Bottom: (Iterated) minimizers of (13) with $\alpha = 8$ and $N = 5, 10, 15$ and 25 .

and $N = 5, 10, 15$ and 25 . In the two left images of Figure 2 it can be realized that the regularization “lags behind” the geometric PDE. This is due to the fact that minimizing (13) produces more *diffuse* solutions; in particular this means that the level sets around the zero isoline are well separated (cf. [14]). Increasing the number of iterations (i.e. decreasing α) decreases this effect. Figure 3 shows the absolute value of the difference between solution $u(t)$ of (1) at time $t = 8$ and (iterated) minimizers of (13) with $\alpha = 8$ ($N = 1$), 4 ($N = 2$), 2 ($N = 4$) and 1 ($N = 8$) (f.l.t.r.). The images are scaled between



Fig. 3. Absolute value of difference: Solution of (1) minus (iterated) minimizers of (13) .

0 (dark) and 1.5 (light), for the image being 150×150 pixels. Thus we might conjecture (what is in fact already motivated by the formal relation between the minimizers of the variational regularization technique and the solution of the MCM-like equation) that for $\alpha \rightarrow 0$ ($N \rightarrow \infty$) the iterated minimizers approximate the solution of the level set equation (12) .

A second experiment is shown in Figure 4, where we used $\alpha = 0.5$ and $\nu = 1$. Note that the initial guess is not simply connected and that topological changes during the evolution are handled automatically. To gain a good

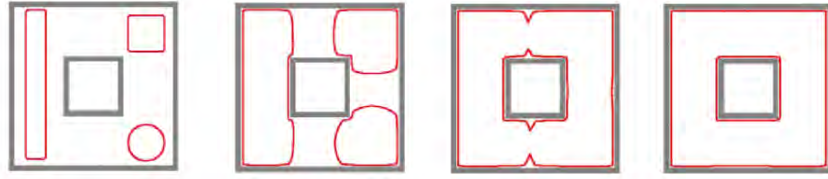


Fig. 4. Initial Image and (iterated) minimizers of (13) with $\alpha = .5$ and $N = 25, 50$ and 75.

reconstruction of non convex structures in an image, one can increase the parameter ν (i.e. g_3). In this case condition (27) implies a smaller value for α in order to guarantee existence of a minimizer. For a further demonstration of the variation regularization technique we use in (13) the setting $g_1 \equiv 1$, $g_2 = g$ with $\varepsilon = 10^{-3}$ and $g_3 \equiv 0$. Figure 5 shows a numerical simulation including changes in the topology of the zero level set. Note that with $g_3 \equiv 0$ no restriction on α has to be imposed.

6 Conclusion

In order to guarantee existence of viscosity solutions of (1) one forces *continuous* initial data $u^0 \in \mathcal{C}^0(\mathbb{R}^2) \cap W^{1,\infty}(\mathbb{R}^2)$ and g sufficiently smooth. (cf. Theorem 1 and 2 above, which have been collected from [7]). In image analysis

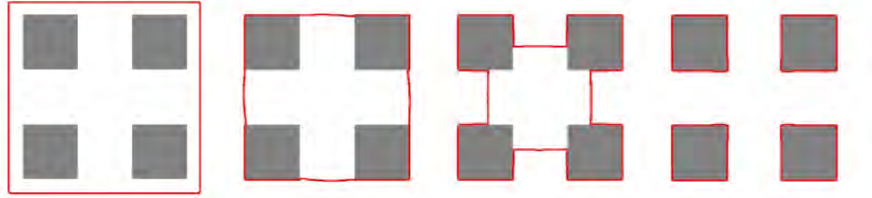


Fig. 5. Initial Image and (iterated) minimizers of (13) with $\alpha = 5$ and $N = 1, 2$ and 3.

the assumption of continuous initial data as well as continuous solutions is not always realistic. The goal of this paper is to show that there exist variational “level set” segmentation techniques, where the analysis allows for discontinuous data and solutions, and moreover, produce comparable numerical results to implicit active contour models.

Acknowledgement. This work has been supported by the Austria Science Foundation (FWF) Projects Y-123INF, FSP 9203-N12 and FSP 9207-N12. For running the algorithms the computer cluster of the HPC - Konsortium Innsbruck was used.

References

1. L. Ambrosio, N. Fusco, and D. Pallara. *Functions of bounded variation and free discontinuity problems*. Oxford Mathematical Monographs. The Clarendon Press Oxford University Press, New York, 2000.
2. G. Bouchitté, I. Fonseca, and L. Mascarenhas. A global method for relaxation. *Arch. Rational Mech. Anal.*, 145(1):51–98, 1998.
3. G. Buttazzo and G. Dal Maso. Γ -limits of integral functionals. *J. Analyse Math.*, 37:145–185, 1980.
4. G. Buttazzo. *Semicontinuity, relaxation and integral representation in the calculus of variation*. Pitman Research Notes in Mathematics. Longman Scientific & Technical, Harlow U.K., 1989.
5. V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numer. Math.*, 66(1):1–31, 1992.
6. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. J. Comput. Vis.*, 22(1):61–79, 1997.
7. V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. Minimal surfaces: A geometric three dimensional segmentation approach. *Numer. Math.*, 77(4):423–451, 1997.
8. Y.-G. Chen, Y. Giga, and S. Goto. Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations. *J. Differ. Geom.*, 33(3):749–786, 1991.
9. M. G. Crandall, H. Ishii, and P.-L. Lions. User’s guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Math. Soc. (N.S.)*, 27(1):1–67, 1992.

10. B. Dacorogna. *Direct methods in the calculus of variations*. Springer-Verlag, Berlin, 1989.
11. L. C. Evans and R. F. Gariepy. *Measure theory and fine properties of functions*. Studies in Advanced Mathematics. CRC Press, Boca Raton, FL, 1992.
12. L. C. Evans and J. Spruck. Motion of level sets by mean curvature. I. *J. Differential Geom.*, 3(3):635–681, 1991.
13. M. Fuchs. Non-convex scale spaces. Diploma thesis, University of Innsbruck, 2005.
14. M. Grasmair, F. Lenzen, A. Obereder, O. Scherzer, and M. Fuchs. A non-convex pde scale space. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 303–315, Berlin, 2005. Springer-Verlag.
15. M. Grasmair and O. Scherzer. Relaxation of nonlocal singular integrals. *Numer. Funct. Anal. Optim.*, 26(4–5):481–506, 2005.
16. F. Guichard and J.-M. Morel. *Image Analysis and P.D.E.'s*. 2000.
17. G. Kühne and J. Weickert. Fast methods for implicit active contour models. *Geometric level set methods in imaging, vision, and graphics*, pages 43–57, 2003.
18. T. Lü, P. Neittaanmäki, and X.-C. Tai. A parallel splitting up method and its application to navier-stokes equations. *Appl. Math. Lett.*, 4(2):25–29, 1991.
19. T. Lü, P. Neittaanmäki, and X.-C. Tai. A parallel splitting-up method for partial differential equations and its applications to navier-stokes equations. *RAIRO Modél. Math. Anal. Numér.*, 26(6):673–708, 1992.
20. S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.
21. E. Radmoser, O. Scherzer, and J. Weickert. Scale-space properties of non-stationary iterative regularization methods. *J. Vis. Comm. Image Represent.*, 11:96–114, 2000.
22. W. Rudin. *Real and complex analysis*. McGraw-Hill Book Co., New York, third edition, 1987.
23. O. Scherzer. Explicit versus implicit relative error regularization on the space of functions of bounded variation. *Inverse problems, image analysis, and medical imaging*, 313:171–198, 2002.
24. O. Scherzer and J. Weickert. Relations between regularization and diffusion filtering. *J. Math. Imaging Vis.*, 12(1):43–63, 2000.
25. R. Temam. *Problèmes mathématique en plasticité*. Méthodes mathématiques de l’informatique. Gauthier-Villars, Montrouge, 1983.
26. J. Weickert. *Anisotropic Diffusion in Image Processing*. ECMI. B.G. Teubner, Stuttgart, 1998.
27. J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, 1998.

Region-Based Variational Problems and Normal Alignment – Geometric Interpretation of Descent PDEs

Jan Erik Solem and Niels Chr. Overgaard

Applied Mathematics Group, School of Technology and Society, Malmö University, Sweden. E-mail: {jes,nco}@ts.mah.se

Summary. Many problems in image analysis and computer vision are formulated using a variational approach. PDE-based methods are often derived from such variational formulations. In this paper a geometric interpretation of these variational problems are explored. In particular the notion of functional gradients and geometric descent directions are defined and discussed. These methods are applied to interesting problems such as region-based segmentation of images, and normal alignment to vector fields. Some new results are derived and some old results reviewed. The presented methodology is illustrated with examples from image analysis.

1 Introduction

A common approach in image analysis and computer vision is to formulate problems in a variational setting. Many inverse problems such as recovering structures (curves, surfaces and regions) from observed data are solved by minimizing “energy” functionals, specifically tailored to the problem at hand. Previously in [19] we have studied a geometric interpretation of variational problems involving m -dimensional surfaces in $\mathbf{R}^{m+1}$, so-called *m-surfaces*. Here we are also going to study a number of region-based problems within the same framework. This type of problems turn up in many important applications such as in e.g., segmentation of images [3, 18]. Other inverse problems such as 3D surface reconstruction [25] and the alignment of curves to image edges [14] are also often formulated as variational problems. Our work is inspired by the variational level set method introduced in [24].

In this paper we clarify some details and try to explain the key ingredients of the gradient interpretation for variational m -surface problems introduced in [19]. A precise geometric definition of descent directions is given. In the case where a geometric gradient of an m -surface functional exists a descent direction can be chosen optimally (as a scalar multiple of the gradient). We also show an example where such an optimal direction does not exist. Instead one

has to resort to “good” descent directions. Furthermore, we analyze region-based functionals, and quotients of these, for which we derive an interesting result concerning the extremals. As examples of this we study the classical Chan-Vese segmentation model [3] as well as modifications of this model. We also show examples of computations for the case of (dense) normal alignment to a vector field, and study gradients and descent directions for this problem. Finally, we relate the geometric framework to the notion of shape gradients [6] used as a computational tool in [1].

2 Background

As a courtesy to the reader, the necessary background on the level set method and the geometric gradient interpretation for variational m -surface problems is briefly recalled here.

2.1 The Kinematics of Dynamic Surfaces

A regular m -surface Γ in $\mathbf{R}^{m+1}$ can be represented implicitly as the zero set of a differentiable function $\phi : \mathbf{R}^{m+1} \rightarrow \mathbf{R}$, the *level set function*, as

$$\Gamma = \{\mathbf{x} : \phi(\mathbf{x}) = 0\} . \quad (1)$$

The sets $\Omega = \{\mathbf{x} : \phi(\mathbf{x}) < 0\}$ and $\{\mathbf{x} : \phi(\mathbf{x}) > 0\}$ are called the *inside* and the *outside* of Γ , respectively. Using this convention, the outward unit normal $\mathbf{n}$ and the mean curvature κ of Γ are given by (cf. [21])

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|} \quad \text{and} \quad \kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} . \quad (2)$$

Other geometric quantities pertaining to Γ , such as the surface area $|\Gamma|$ and the volume $|\Omega|$ of Ω , can also be expressed in terms of ϕ . Clearly

$$|\Omega| = \int_{\Omega} d\mathbf{x} = \int (1 - H(\phi)) d\mathbf{x} , \quad (3)$$

where $H(\cdot)$ is the Heaviside function, and by using a well-known result from distribution theory (see e.g. Hörmander [10, Thm. 6.1.5]), the Euclidean surface measure on Γ can be expressed as

$$d\sigma = |\nabla \phi| \delta(\phi) d\mathbf{x} . \quad (4)$$

Here $\delta \in \mathcal{D}'(\mathbf{R})$ is the Dirac distribution on the real line, and $\delta(\phi) \in \mathcal{D}'(\mathbf{R}^{m+1})$ denotes the *pullback* of δ by ϕ . (That is, the composition of δ by the function ϕ .) It follows from (4) that

$$|\Gamma| = \int_{\Gamma} d\sigma = \int |\nabla \phi| \delta(\phi) d\mathbf{x} = \int |\nabla H(\phi)| d\mathbf{x} ,$$

the last equality being a special case of the *co-area formula*¹.

The implicit representation introduced above can be used to define a *dynamic surface* (or *surface evolution*), $t \mapsto \Gamma(t)$, by adding a time dependence to the level set function: $\phi = \phi(\mathbf{x}, t)$. The dynamic surface is then given by

$$t \mapsto \Gamma(t) := \{\mathbf{x} : \phi(\mathbf{x}, t) = 0\} . \quad (5)$$

We now want to introduce the notion of the *normal velocity* of a surface evolution (5). The normal velocity is going to be a real-valued function $v = v(t, \mathbf{x})$ defined on the surface $\Gamma(t)$. We recall from [19] the following argument which motivates the definition given below. Suppose a particle moves along with the dynamic surface $\Gamma(t)$. If the motion of the particle is described by the parameterized curve $t \mapsto \boldsymbol{\alpha}(t)$ with $\boldsymbol{\alpha}(0) = \mathbf{x}_0 \in \Gamma$, then the equality $\phi(\boldsymbol{\alpha}(t), t) = 0$ holds identically at all times t . Differentiation of this identity gives

$$\dot{\boldsymbol{\alpha}}(0) \cdot \mathbf{n} = -\frac{\partial \phi(\mathbf{x}_0, 0)/\partial t}{|\nabla \phi(\mathbf{x}_0, 0)|} , \quad (6)$$

The left-hand side is the normal component of the velocity $\dot{\boldsymbol{\alpha}}(0)$ of the particle at $t = 0$. The normal component is an intrinsic property of the evolution because it is independent of the particular choice of the curve $\boldsymbol{\alpha}$ and the level set function $\phi(\mathbf{x}, t)$, cf. [19]. We therefore define the *normal velocity* of the evolution $\Gamma(t)$ as the function

$$\dot{\Gamma}(t, \mathbf{x}) = -\frac{\partial \phi(\mathbf{x}, t)/\partial t}{|\nabla \phi(\mathbf{x}, t)|} \quad (\mathbf{x} \in \Gamma(t)) . \quad (7)$$

Using the notation $v(\Gamma) = \dot{\Gamma}(t)$ we can rewrite this equation as

$$\frac{\partial \phi}{\partial t} + v(\Gamma)|\nabla \phi| = 0 , \quad (8)$$

where we have dropped the dependence on $\mathbf{x}$ and t to simplify the notation. This is the well-known *level set equation* which is the basis for the level set method, introduced independently by [7] and [16] as a tool for evolving implicit surfaces.

¹In fact, (4) is, at least formally, an infinitesimal version of the co-area formula: If the Euclidean surface measure on the set $\Gamma_t = \{\mathbf{x} : \phi(\mathbf{x}) = t\}$ is denoted by $d\sigma_t$, then

$$d\sigma_t = |\nabla \phi| \delta(\phi - t) d\mathbf{x} .$$

Suppose now that $f(t) \in C_0(\mathbf{R})$ and $\psi(\mathbf{x}) \in C_0(\mathbf{R}^{m+1})$ (continuous functions with compact support) then

$$\begin{aligned} \int_{-\infty}^{\infty} f(t) \left(\int_{\Gamma_t} \psi(\mathbf{x}) d\sigma_t \right) dt &= \int_{-\infty}^{\infty} f(t) \int \psi(\mathbf{x}) |\nabla \phi| \delta(\phi - t) d\mathbf{x} dt \\ &= \int \psi(\mathbf{x}) |\nabla \phi| \left(\int_{-\infty}^{\infty} f(t) \delta(\phi - t) dt \right) d\mathbf{x} = \int \psi(\mathbf{x}) f(\phi) |\nabla \phi| d\mathbf{x} , \end{aligned}$$

which is the co-area formula. To simplify notation, the $\mathbf{x}$ in $\phi(\mathbf{x})$ was omitted.

2.2 Geometric Gradient Descent for Dynamic Surfaces

In this section we recall from [19] the construction of gradient descent evolutions for the minimization of functionals $E(\Gamma)$ defined on manifolds of admissible m -surfaces Γ .

Let us imagine that the set of admissible m -surfaces constitutes an infinite-dimensional manifold M . Then each admissible m -surface Γ is considered as a “point” on M . At $\Gamma \in M$ the *tangent space* $T_\Gamma M$ is the set of all functions $v : \Gamma \rightarrow \mathbf{R}$ such that v corresponds to the normal velocity $\dot{\Gamma}(0)$ of some surface evolution $t \mapsto \Gamma(t)$ with $\Gamma(0) = \Gamma$. Each tangent space $T_\Gamma M$ of M is endowed with a scalar product $\langle \cdot, \cdot \rangle_\Gamma$ defined by the integral

$$\langle v, w \rangle_\Gamma = \int_\Gamma v(\mathbf{x})w(\mathbf{x}) d\sigma \quad (v, w \in T_\Gamma M) . \quad (9)$$

If the norm of $v \in T_\Gamma M$ is defined by $\|v\|_\Gamma = \sqrt{\langle v, v \rangle_\Gamma}$, then we have Schwarz’ inequality:

$$|\langle v, w \rangle_\Gamma| \leq \|v\|_\Gamma \|w\|_\Gamma \quad (v, w \in T_\Gamma M) . \quad (10)$$

Now, consider a functional $E : M \rightarrow \mathbf{R}$ and let $\Gamma \in M$ be fixed. E is said to be Gâteaux-differentiable at Γ , if the derivative

$$dE(\Gamma)v = \left. \frac{d}{dt} E(\Gamma(t)) \right|_{t=0} \quad (11)$$

exists for every $v \in T_\Gamma M$. Here $\Gamma(t)$ is any surface evolution satisfying $\Gamma(0) = \Gamma$ and $\dot{\Gamma}(0) = v$. The functional $dE(\Gamma)$ defined in (11) is homogeneous of degree one in its argument, but not necessarily additive (and consequently not linear). If E is such that the right hand side of (11) is a linear functional, then E is said to be differentiable at Γ and $dE(\Gamma)$ is called the Gâteaux derivative (or the functional derivative, or the differential) of E at Γ . There sometimes exists a vector $\nabla E(\Gamma) \in T_\Gamma M$ such that the following identity holds for all normal velocities $v \in T_\Gamma M$:

$$dE(\Gamma)v = \langle \nabla E(\Gamma), v \rangle_\Gamma \quad (\text{Riesz}) . \quad (12)$$

If this is the case, then $\nabla E(\Gamma)$ is called the L^2 -gradient of E at Γ , and it is uniquely determined by the property (12)².

The *gradient descent* for the variational problem $\min_\Gamma E(\Gamma)$ is now defined by the following initial value problem

$$\dot{\Gamma}(t) = -\nabla E(\Gamma(t)); \quad \Gamma(0) = \Gamma_0 , \quad (13)$$

²It would be more correct to use the notation $\nabla_M E$ for the gradient of E , as it is actually the intrinsic gradient of E on the manifold M of admissible m -surfaces. In this paper, functionals on M are always denoted by upper case letters, so it should not cause any confusion to use the abbreviated notation ∇E .

where Γ_0 is some initial m -surface. Recently researchers have started looking at using different inner products, leading to other gradients than the L^2 -gradient above, cf. e.g., [4, 23]. Droske and Rumpf [8, §3] also mention that the gradient descent motion for a surface functional can be defined in terms of the L^2 -gradient. However, their aim is to find a formulation in which all the level sets of ϕ are evolved simultaneously. They achieve this by defining a global energy $\phi \mapsto E[\phi]$, which is the weighted sum of the surface functional on each level set $\{\phi = c\}$, and by defining a (Riemannian) metric on the linear space of functions ϕ . Here on the contrary, we are interested in following a single level set surface and in performing analytic computations directly on the implicit representation.

We end this section with a very useful result concerning the gradient of a general class of functionals defined on m -surfaces,

$$E(\Gamma) = \int_{\Gamma} g(\mathbf{x}, \mathbf{n}) \, d\sigma, \quad (14)$$

where $g : \mathbf{R}^{m+1} \times S^m \rightarrow \mathbf{R}$ is a function of both position and surface orientation. For functionals of this type the following holds.

Theorem 1. *Let $g = g(\mathbf{x}, \mathbf{n})$ be continuously differentiable with respect to $\mathbf{x}$ and twice continuously differentiable with respect to $\mathbf{n}$. Then the functional E defined by (14) has the differential*

$$dE(\Gamma)v = \langle \nabla \cdot [\nabla_{S^m} g + g \mathbf{n}], v \rangle_{\Gamma},$$

for $v \in T_{\Gamma}M$. In particular $\nabla E = \nabla \cdot [\nabla_{S^m} g + g \mathbf{n}]$.

This result was proved in [9], using Cartan's method of moving frames, and (independently) by the authors in [19]. The proof we are going to give here is an abbreviated version of the one given in [19]. It illustrates how the Gâteaux derivatives of surface functionals can be effectively calculated by operating directly in the level set formulation, using a bit of distribution theory.

Proof. Consider the surface evolution $s \mapsto \Gamma(s) = \{\mathbf{x} : \phi^s(\mathbf{x}) = 0\}$ given by the variation $\phi^s = \phi + s\psi$ of ϕ . By (7) the corresponding normal velocity at $s = 0$ is $v = -\psi/|\nabla\phi|$. Taking the Gâteaux derivative with $v = -\psi/|\nabla\phi|$ gives

$$dE(\Gamma)v = \frac{d}{ds} E(\phi + s\psi) \Big|_{s=0} = \int \frac{d}{ds} \left[g \left(\mathbf{x}, \frac{\nabla\phi^s}{|\nabla\phi^s|} \right) |\nabla\phi^s| \delta(\phi^s) \right] d\mathbf{x} \Big|_{s=0}.$$

Let us use the notation $\mathbf{g}_{\mathbf{n}} = \nabla_{S^m} g$ for the gradient on the unit sphere S^m . Then clearly $\mathbf{g}_{\mathbf{n}} \in T_{\mathbf{n}}S^m$, hence $\mathbf{g}_{\mathbf{n}} \cdot \mathbf{n} = 0$. This means that the derivative $\frac{d}{ds} g \left(\mathbf{x}, \frac{\nabla\phi^s}{|\nabla\phi^s|} \right)$ is

$$\frac{d}{ds} g \left(\mathbf{x}, \frac{\nabla\phi^s}{|\nabla\phi^s|} \right) = \mathbf{g}_{\mathbf{n}} \cdot \left[\frac{\nabla\psi}{|\nabla\phi|} - \left(\frac{\nabla\phi}{|\nabla\phi|} \cdot \frac{\nabla\psi}{|\nabla\phi|} \right) \frac{\nabla\phi}{|\nabla\phi|} \right] = \mathbf{g}_{\mathbf{n}} \cdot \frac{\nabla\psi}{|\nabla\phi|}.$$

The Gâteaux derivative is then simply

$$dE(\Gamma)v = \int g_{\mathbf{n}} \cdot \nabla \psi \delta(\phi) d\mathbf{x} + \int g \left[\frac{\nabla \phi}{|\nabla \phi|} \cdot \nabla \psi \right] \delta(\phi) d\mathbf{x} + \int g |\nabla \phi| \delta'(\phi) \psi d\mathbf{x} .$$

Integration by parts on $\nabla \psi$ gives

$$\begin{aligned} dE(\Gamma)v &= \int (-\psi) \nabla \cdot [g_{\mathbf{n}} \delta(\phi)] d\mathbf{x} + \int (-\psi) \nabla \cdot \left[g \frac{\nabla \phi}{|\nabla \phi|} \delta(\phi) \right] d\mathbf{x} \\ &\quad - \int (-\psi) g |\nabla \phi| \delta'(\phi) d\mathbf{x} \\ &= \int (-\psi) \left\{ (\nabla \cdot \nabla_{S^m} g) \delta(\phi) + g_{\mathbf{n}} \cdot \nabla \phi \delta'(\phi) + \nabla \cdot \left[g \frac{\nabla \phi}{|\nabla \phi|} \right] \delta(\phi) \right. \\ &\quad \left. + g \frac{\nabla \phi}{|\nabla \phi|} \cdot \nabla \phi \delta'(\phi) - g |\nabla \phi| \delta'(\phi) \right\} d\mathbf{x} . \end{aligned}$$

Since $g_{\mathbf{n}} \cdot \mathbf{n} = 0$ implies $g_{\mathbf{n}} \cdot \nabla \phi = 0$ and the two last terms cancel we get

$$\begin{aligned} dE(\Gamma)v &= \int \left(\frac{-\psi}{|\nabla \phi|} \right) \nabla \cdot \left[g_{\mathbf{n}} + g \frac{\nabla \phi}{|\nabla \phi|} \right] |\nabla \phi| \delta(\phi) d\mathbf{x} \\ &= \langle v, \nabla \cdot [g_{\mathbf{n}} + g \mathbf{n}] \rangle_{\Gamma} \\ &= \langle v, \nabla \cdot [\nabla_{S^m} g + g \mathbf{n}] \rangle_{\Gamma} , \end{aligned}$$

which is the desired result. $\square$

The gradient descent evolution for the minimization of (14), when formulated in terms of a level set function ϕ , is

$$\frac{\partial \phi}{\partial t} = (\nabla \cdot [\nabla_{S^m} g + g \mathbf{n}]) |\nabla \phi| .$$

This follows from our definition of gradient descent (13), the fact that the normal velocity of the evolution $\Gamma(t)$, represented by $\phi(\mathbf{x}, t)$, is $\dot{\Gamma}(t) = -(\partial \phi / \partial t) / |\nabla \phi|$, and that $-\nabla E = -\nabla \cdot [\nabla_{S^m} g + g \mathbf{n}]$.

3 Descent Directions

In this section we will define what we mean by descent directions for a given functional $E(\Gamma)$. Descent directions are used to define an m -surface evolution as a procedure for minimizing E .

One common method of deriving the surface evolution is through the Euler-Lagrange equation which one obtains by setting the first variation equal to zero for all perturbations of ϕ . For functionals of the form (14) this results in an expression of the form $G(\mathbf{x}, \phi) \delta(\phi) = 0$. The surface evolution is then obtained by solving

$$\frac{\partial \phi}{\partial t} = G(\mathbf{x}, \phi) \delta(\phi) , \quad (15)$$

until a steady state is reached. Often this equation is replaced by

$$\frac{\partial \phi}{\partial t} = G(\mathbf{x}, \phi) |\nabla \phi| , \quad (16)$$

or

$$\frac{\partial \phi}{\partial t} = G(\mathbf{x}, \phi) , \quad (17)$$

cf. e.g., [20], where perhaps (16) is the most common alternative.

The notation in the literature varies and sometimes “descent direction” is taken to mean either the entire PDE, the right-hand-side or the function $G(\mathbf{x}, \phi)$. We are interested in building a geometric framework and as stated above, the normal velocity is an intrinsic property of any surface evolution. Therefore, there is only one alternative for defining descent directions in a geometric manner. Let us state precisely what we mean by a descent direction.

Definition 1. *A descent direction for a differentiable functional E is a normal velocity $v \in T_\Gamma M$ which satisfies $dE(\Gamma)v \leq 0$.*

When the gradient ∇E exists the condition in the definition is equivalent to

$$\langle \nabla E, v \rangle_\Gamma \leq 0 . \quad (18)$$

This latter condition is easily checked since it does not require the computation of any Gâteaux derivatives, one simply evaluates (18). The functional defined by $\langle \nabla E, v \rangle_\Gamma$ is a continuous linear bounded functional on $T_\Gamma M$.

If ∇E exists it is also interesting to note that since we have an inner product, there is an optimal descent direction (up to multiplication with a scalar³), namely $v = -\nabla E$. Also, it is possible to compare directions since if $\langle \nabla E, v \rangle_\Gamma < \langle \nabla E, w \rangle_\Gamma$ for $v, w \in T_\Gamma M$, then v is “better” than w .

4 Region-Based Functionals

For many applications one is interested in finding regions (where the boundary is given by an m -surface) in sets of measured data, such as images and MRI data. One example is image segmentation where the goal is to partition an image into different regions. In the same way as functionals were defined on a surface Γ in (14), one can formulate region-based functionals where Γ is the (unknown) boundary of some region to be found.

Let $\Omega := \text{int}(\Gamma)$ denote the interior of Γ , $f(\mathbf{x}) : \mathbf{R}^{m+1} \rightarrow \mathbf{R}$ be a given potential function, and define a functional as the volume integral

³i.e., there is an optimal direction v with a given length $\|v\|_\Gamma$.

$$E(\Gamma) = \int_{\Omega} f(\mathbf{x}) \, d\mathbf{x} . \quad (19)$$

This is a general form of a region-based functional where f can be e.g., the deviation from an image model [3, 18].

When the region Ω is perturbed, only the change at the boundary Γ will affect the value of E . It is therefore not surprising that (19) has a gradient interpretation in the sense of Section 2.2. The differential of (19) is a well-known classical result, cf. e.g., [18]. We state the differential and the corresponding gradient in a theorem.

Theorem 2. *The gradient of $E(\Gamma)$ in (19) is $\nabla E(\Gamma) = f(\mathbf{x})$.*

Proof. The Gâteaux derivative of E is

$$\begin{aligned} dE(\Gamma)v &= \left. \frac{d}{ds} E(\phi + s\psi) \right|_{s=0} = \int \left. \frac{d}{ds} (1 - H(\phi + s\psi)) \right|_{s=0} f(\mathbf{x}) \, d\mathbf{x} \\ &= \int (-\psi)\delta(\phi)f(\mathbf{x}) \, d\mathbf{x} = \int \left(\frac{-\psi}{|\nabla\phi|} \right) f(\mathbf{x}) |\nabla\phi| \delta(\phi) \, d\mathbf{x} \\ &= \int_{\Gamma} v f(\mathbf{x}) \, d\sigma = \langle v, f(\mathbf{x}) \rangle_{\Gamma} , \end{aligned} \quad (20)$$

where (3) and (4) were used. So the result follows from (12). $\square$

The gradient descent motion for minimizing E is $\partial\phi/\partial t = f(\mathbf{x})|\nabla\phi|$, where the normal velocity is $v = -f(\mathbf{x})$.

4.1 The Chan-Vese Model

In traditional active contour methods, such as snakes [12], geometric active contours [2], and other similar models [5, 22], the segmentation of an image is achieved by evolving a simple, closed, parametrized curve (the active contour), using a driving force provided by an edge map constructed from the original image. The evolution is continued until an equilibrium configuration is reached close to the edges in the images.

Most edge maps are based on image gradients, and therefore require images in which the edges are rather distinct. However, many images exist in which the “edges” are too diffuse to be adequately captured by any edge map construction. To deal with such images Chan and Vese suggested, in the by now classic paper [3], to use a driving force derived from an area-based “energy”, which uses global image information. Moreover, they formulated the theory in the level set framework in order to cope with topological changes.

Let us briefly recall the Chan-Vese model which is inspired by the classical work of Mumford and Shah [15]. Let $I = I(\mathbf{x}) : D \rightarrow \mathbf{R}$ denote the image to be segmented, $D \subset \mathbf{R}^2$ being the image domain. Also, let Γ denote a simple closed curve in the image domain (or a non-overlapping union of such

curves, bearing in mind that this is allowed in the level set framework), and set $\Omega_0 = \Omega_0(\Gamma) := \text{int}(\Gamma)$ and $\Omega_1 = \Omega_1(\Gamma) := \text{ext}(\Gamma)$. Consider the functional:

$$E(\boldsymbol{\mu}, \Gamma) = \frac{1}{2} \int_{\Omega_0} |I(\mathbf{x}) - \mu_0|^2 d\mathbf{x} + \frac{1}{2} \int_{\Omega_1} |I(\mathbf{x}) - \mu_1|^2 d\mathbf{x} + \alpha |\Gamma|, \quad (21)$$

where $\boldsymbol{\mu} = (\mu_0, \mu_1) \in \mathbf{R}^2$ is a pair of parameters, $|\Gamma|$ denotes the length of the curve Γ , and $\alpha > 0$ is a fixed weight. The idea of the method presented in [3] is to find a curve Γ^* and a pair of parameters $\boldsymbol{\mu}^*$ which solves the optimization problem,

$$E(\boldsymbol{\mu}^*, \Gamma^*) = \min_{\boldsymbol{\mu}, \Gamma} E(\boldsymbol{\mu}, \Gamma). \quad (22)$$

The segmentation of the image I is defined as the partition of the image domain induced by the optimal curve Γ^* . This partition is found using gradient descent on Γ where the gradient is

$$\nabla E = \frac{1}{2}(I(\mathbf{x}) - \mu_0)^2 - \frac{1}{2}(I(\mathbf{x}) - \mu_1)^2 + \alpha \kappa,$$

from Theorem 2.

In the remainder of this section we are going to discuss a slightly generalized version of the Chan-Vese model, proposed in [11], in which the quadratic penalty function $\frac{1}{2}(\cdot)^2$ is replaced by a more general penalty function $V(\cdot)$:

$$E(\boldsymbol{\mu}, \Gamma) = \int_{\Omega_0} V(I(\mathbf{x}) - \mu_0) d\mathbf{x} + \int_{\Omega_1} V(I(\mathbf{x}) - \mu_1) d\mathbf{x} + \alpha |\Gamma|. \quad (23)$$

In the following we shall require that V is a strictly convex function and that $V(t) \rightarrow \infty$ as $|t| \rightarrow \infty$. A feasible choice, which gives a natural generalization of (21), is obtained by taking the penalizer V to be one of the functions

$$V(t) = \frac{1}{p} \langle t \rangle_\epsilon^p, \quad 1 \leq p < \infty, \epsilon \geq 0, \quad (24)$$

where $\langle t \rangle_\epsilon = \sqrt{t^2 + \epsilon^2}$ is a regularization of the absolute value $|t|$. If $p = 1$ we require $\epsilon > 0$ in (24) in order that V be strictly convex. The case of using the L^1 -norm in the fidelity term, i.e., $p = 1$ and $\epsilon = 0$, was mentioned in [13]. Notice that for $p = 2$ and $\epsilon = 0$ the Chan-Vese functional (21) is recovered. It is sometimes desirable to use penalty functions of the form (24) with $1 \leq p < 2$ because the resulting segmentation models will be more robust with respect to noise and outliers in the data.

In order to solve the optimization problem (22) for the generalized functional (23), we begin by “separating” the variables in the minimization

$$\min_{\boldsymbol{\mu}, \Gamma} E(\boldsymbol{\mu}, \Gamma) = \min_{\Gamma} \left[\min_{\boldsymbol{\mu}} E(\boldsymbol{\mu}, \Gamma) \right],$$

that is, for Γ fixed we determine the optimal parameters $\boldsymbol{\mu} = \boldsymbol{\mu}(\Gamma)$, then we try to find the optimal contour Γ^* by minimizing the *reduced functional*

$\hat{E}(\Gamma) := E(\boldsymbol{\mu}(\Gamma), \Gamma)$, in which case the corresponding optimal parameter in (22) is $\boldsymbol{\mu}^* = \boldsymbol{\mu}(\Gamma^*)$.

Now, for the Chan-Vese functional it is easy to find the optimal parameters for each fixed Γ ; they are simply the mean intensities of the image taken over each of the sub-domains cut out by Γ ,

$$\mu_i(\Gamma) = \frac{1}{|\Omega_i|} \int_{\Omega_i} I(\mathbf{x}) d\mathbf{x}, \quad (i = 0, 1), \quad (25)$$

where $|\Omega_i|$ denotes the area of the set $\Omega_i \subset \mathbf{R}^2$. For other choices of the penalty function V we can in general not expect to find such nice explicit formulas. However, when V is strictly convex, and $V(t) \rightarrow \infty$ as $|t| \rightarrow \infty$, each of the functions

$$\mu_i \mapsto \int_{\Omega_i} V(I(\mathbf{x}) - \mu_i) d\mathbf{x}, \quad (i = 0, 1),$$

is also strictly convex and tends to infinity as $|\mu_i| \rightarrow \infty$. It therefore follows from standard optimization theory that there exists a unique pair of optimal parameters $\mu_i = \mu_i(\Gamma)$ for each fixed curve Γ . These optimal parameters can be computed using Newton's method or some other method from optimization theory.

Having ascertained the existence of a unique pair of optimal parameters $\boldsymbol{\mu}(\Gamma)$ for each Γ , it remains to find the optimal curve Γ^* . We use gradient descent, so we have to compute the Gâteaux derivative of the reduced functional $\hat{E}(\Gamma) := E(\boldsymbol{\mu}(\Gamma), \Gamma)$. By the chain rule,

$$d\hat{E}(\Gamma)v = \frac{\partial}{\partial \boldsymbol{\mu}} E(\boldsymbol{\mu}(\Gamma), \Gamma) \cdot d\boldsymbol{\mu}(\Gamma)v + dE(\boldsymbol{\mu}, \Gamma)v.$$

At first it seems as if we have to compute the Gâteaux derivative $d\boldsymbol{\mu}(\Gamma)$, which could be complicated in view of the fact that we have no explicit formula for $\boldsymbol{\mu}(\Gamma)$. However, as observed in [11], $(\partial/\partial \boldsymbol{\mu})E(\boldsymbol{\mu}(\Gamma), \Gamma) = 0$ because $\boldsymbol{\mu}(\Gamma)$ minimizes E for Γ fixed. Using this fact, the derivative of the reduced functional is simply

$$d\hat{E}(\Gamma)v = dE(\boldsymbol{\mu}(\Gamma), \Gamma)v = \int_{\Gamma} \left[V(I(\mathbf{x}) - \mu_0(\Gamma)) - V(I(\mathbf{x}) - \mu_1(\Gamma)) \right] v d\sigma \quad (26)$$

for any normal velocity v on Γ . That is, the derivative of the reduced functional $\hat{E}(\Gamma)$ is the derivative of $E(\boldsymbol{\mu}(\Gamma), \Gamma)$ computed as if $\boldsymbol{\mu}(\Gamma)$ is a constant, independent of Γ . It follows that the gradient of the generalized Chan-Vese functional is $\nabla E = V(I(\mathbf{x}) - \mu_0(\Gamma)) - V(I(\mathbf{x}) - \mu_1(\Gamma)) + \alpha\kappa$, so the corresponding gradient descent evolution $t \mapsto \Gamma(t)$, formulated in the level set framework, becomes

$$\frac{\partial \phi}{\partial t} = \left[V(I(\mathbf{x}) - \mu_0(\Gamma)) - V(I(\mathbf{x}) - \mu_1(\Gamma)) + \alpha\kappa \right] |\nabla \phi|, \quad (27)$$

where the level set function $\phi = \phi(\mathbf{x}, t)$ is chosen such that $\Omega_0(t) := \text{int}(\Gamma(t)) = \{\mathbf{x} : \phi(\mathbf{x}, t) < 0\}$.

Example. The experiment shown in Figure 1 compares the use of the penalty functions $V(t) = \frac{1}{2}t^2$ (the original Chan-Vese model) and $V(t) = \langle t \rangle_{0.5}$ (i.e. (24) with $p = 1$ and $\epsilon = 0.5$). The synthetic test image consists of a bright square of intensity 125 on a dark background with intensity 75. Inside the square, $1/8$ of the pixels have been randomly chosen, and their value set to zero. Likewise, $1/8$ of the background-pixels have been randomly chosen and their value set to 200. For the Chan-Vese model, $\alpha = 1500$ was chosen, and for the sub-quadratic model, $\alpha = 70$. (These choices have been judged nearly optimal for each of the models.) The sub-quadratic model converges faster to the desired contour than the Chan-Vese model. Moreover, the latter tends to over-segment the image in the sense that it picks up noise during the evolution.

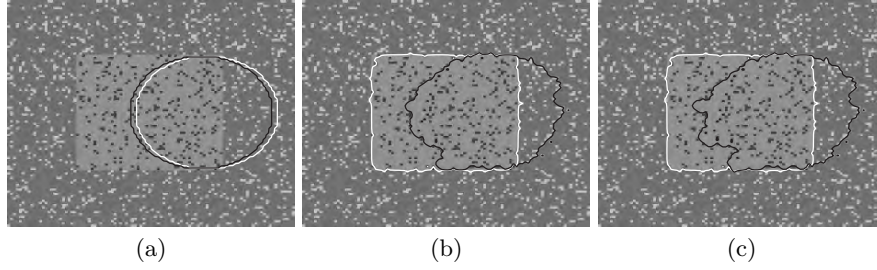


Fig. 1. Comparison between the Chan-Vese model (black) and a generalized Chan-Vese model (white) using the penalty function $V(t) = \frac{1}{p}\langle t \rangle_{\epsilon}^p$ with $p = 1$ and $\epsilon = 0.5$. The test image is a bright square on a dark background with randomly added outliers. (a) initial curve, (b) after 600 iterations, (c) after 2000 iterations. The generalized model is seen to converge much faster to the desired contour.

4.2 Quotients of Region-Functionals

In some applications it can be desirable to minimize the *average* of a potential, $f : \mathbf{R}^{m+1} \rightarrow \mathbf{R}$, inside Γ instead of just minimizing the integral of f as in (19). One reason for this is that the region-functional (19) depends on $|\Omega|$, the volume of the set Ω . This means that smaller Ω are preferred and if $f \geq 0$, the global minimum is $\Omega = \emptyset$, an empty region. The functional representing the average potential is

$$E(\Gamma) = \frac{\int_{\Omega} f(\mathbf{x}) d\mathbf{x}}{\int_{\Omega} d\mathbf{x}} = \frac{1}{|\Omega|} \int_{\Omega} f(\mathbf{x}) d\mathbf{x} . \quad (28)$$

For this particular case we have the following result:

Proposition 1. *Extremals of the functional (28) are level sets of $f(\mathbf{x})$.*

Proof. Using the standard quotient rule, the Gâteaux derivative of E is

$$\begin{aligned} dE(\Gamma)v &= \frac{(\int_{\Omega} d\mathbf{x}) (\int_{\Gamma} f(\mathbf{x})v d\sigma) - (\int_{\Omega} f(\mathbf{x}) d\mathbf{x}) (\int_{\Gamma} v d\sigma)}{(\int_{\Omega} d\mathbf{x})^2} \\ &= \frac{\int_{\Gamma} f(\mathbf{x})v d\sigma - E(\Gamma) \int_{\Gamma} v d\sigma}{\int_{\Omega} d\mathbf{x}} = \frac{1}{|\Omega|} \int_{\Gamma} [f(\mathbf{x}) - E(\Gamma)]v d\sigma . \end{aligned} \quad (29)$$

At an extremal of $E(\Gamma)$

$$\int_{\Gamma} [f(\mathbf{x}) - E(\Gamma)]v d\sigma = 0 ,$$

holds for all normal velocities v since $E(\Gamma)$ is constant and $|\Omega| > 0$. From this relation it follows that $f(\mathbf{x})$ is constant at extremals. $\square$

From this simple calculation we can state the more general result:

Corollary 1. *Extremals of the functional*

$$E(\Gamma) = \frac{\int_{\Omega} f(\mathbf{x}) d\mathbf{x}}{\int_{\Omega} g(\mathbf{x}) d\mathbf{x}} ,$$

are level sets of the function $f(\mathbf{x})/g(\mathbf{x})$.

Proof. The proof follows from the same calculation as for the proposition above. We leave the details to the reader. $\square$

From the Gâteaux derivative (29) we find that the gradient of the functional E defined in (28) is

$$\nabla E(\Gamma) = \frac{f(\mathbf{x}) - E(\Gamma)}{|\Omega|} ,$$

and a useful descent direction (in the sense of Section 3) is

$$v = -[f(\mathbf{x}) - E(\Gamma)] . \quad (30)$$

Example. Figure 2 shows an example of using the the descent direction (30) for the region quotient functional (28) with $f(\mathbf{x}) = I(\mathbf{x})$, the image gray level value. The curve is simply moved with a normal velocity that only depends on the average inside Γ and the local gray level value. Note that the evolution does not depend on the “shape”, such as the curvature, of the curve. The image size is 100×100 .

4.3 Further Comments On Region-Based Segmentation

In [1, Sec. 2.3] the following segmentation model was considered:

$$\tilde{E}(\Gamma) = E(\tilde{\mu}(\Gamma), \Gamma), \quad (31)$$



Fig. 2. An example illustrating curve evolution with normal velocity given by (30). (a) initial curve, (b) after 40 iterations, (c) after 70 iterations, (d) final curve after 100 iterations.

where $E(\boldsymbol{\mu}, \Gamma)$ is the generalized Chan-Vese functional (23) studied in Section 4.1, and $\tilde{\boldsymbol{\mu}}(\Gamma) = (\tilde{\mu}_0(\Gamma), \tilde{\mu}_1(\Gamma))$ is determined by:

$$\tilde{\mu}_i(\Gamma) = \frac{1}{|\Omega_i|} \int_{\Omega_i} I(\mathbf{x}) d\mathbf{x}, \quad (i = 0, 1). \quad (32)$$

Observe that instead of defining $\tilde{\boldsymbol{\mu}}(\Gamma)$ as the parameter pair minimizing $\boldsymbol{\mu} \mapsto E(\boldsymbol{\mu}, \Gamma)$ for Γ fixed, as in the generalized Chan-Vese model, the authors of [1] insist that the parameters should be the mean intensity of the image over each sub-domain defined by Γ . This choice of $\tilde{\boldsymbol{\mu}}$ is motivated by the wish to use classical statistical quantities to characterize the properties of the image.

Again, to find an admissible curve Γ^* such that

$$\tilde{E}(\Gamma^*) = \min_{\Gamma} \tilde{E}(\Gamma)$$

we resort to the gradient descent method, hence we must compute the Gâteaux derivative of the functional $\tilde{E}$. Using the chain rule we find that

$$d\tilde{E}(\Gamma)v = \frac{\partial}{\partial \boldsymbol{\mu}} E(\tilde{\boldsymbol{\mu}}(\Gamma), \Gamma) \cdot d\tilde{\boldsymbol{\mu}}(\Gamma)v + dE(\tilde{\boldsymbol{\mu}}(\Gamma), \Gamma)v.$$

This time around, the partial derivatives

$$\frac{\partial}{\partial \mu_i} E(\tilde{\boldsymbol{\mu}}(\Gamma), \Gamma) = - \int_{\Omega_i} V'(I(\mathbf{x}) - \tilde{\mu}_i) d\mathbf{x} \quad (i = 0, 1)$$

are not necessarily zero, as was the case for the generalized Chan-Vese model. Instead we have to compute the Gâteaux derivatives of the quotients in (32) defining the mean intensities. Using (29) we easily see that

$$d\tilde{\mu}_i(\Gamma)v = \frac{1}{|\Omega_i|} \int_{\Omega_i} [I(\mathbf{x}) - \tilde{\mu}_i(\Gamma)]v d\mathbf{x}, \quad (i = 0, 1). \quad (33)$$

Since we know that (see Equation (26))

$$dE(\tilde{\boldsymbol{\mu}}(\Gamma), \Gamma)v = \int_{\Gamma} \left[V(I(\mathbf{x}) - \tilde{\mu}_0(\Gamma)) - V(I(\mathbf{x}) - \tilde{\mu}_1(\Gamma)) \right] v d\mathbf{x},$$

the gradient of $\tilde{E}$ becomes (cf. [1, Sec. 5.4])

$$\begin{aligned} \nabla \tilde{E}(\Gamma) &= \frac{\tilde{\mu}_0(\Gamma) - I(\mathbf{x})}{|\Omega_0|} \int_{\Omega_0} V'(I(\mathbf{x}) - \tilde{\mu}_0(\Gamma)) d\mathbf{x} \\ &\quad - \frac{\tilde{\mu}_1(\Gamma) - I(\mathbf{x})}{|\Omega_1|} \int_{\Omega_1} V'(I(\mathbf{x}) - \tilde{\mu}_1(\Gamma)) d\mathbf{x} \\ &\quad + \left[V(I(\mathbf{x}) - \tilde{\mu}_0(\Gamma)) - V(I(\mathbf{x}) - \tilde{\mu}_1(\Gamma)) \right]. \end{aligned} \quad (34)$$

Compared to the gradient $\nabla \hat{E}(\Gamma) = V(I(\mathbf{x}) - \mu_0(\Gamma)) - V(I(\mathbf{x}) - \mu_1(\Gamma))$ of the reduced functional for the generalized Chan-Vese model, the above gradient contains two extra terms due to the fact that the parameter pair $\tilde{\boldsymbol{\mu}}(\Gamma)$ does not necessarily minimize $E(\boldsymbol{\mu}, \Gamma)$. Observe that if $V(t) = \frac{1}{2}t^2$, then the above model coincides with the Chan-Vese model. In fact, $\int_{\Omega_i} V'(I(\mathbf{x}) - \tilde{\mu}_i(\Gamma)) d\mathbf{x} = \int_{\Omega_i} I(\mathbf{x}) - \tilde{\mu}_i(\Gamma) d\mathbf{x} = 0$, ($i = 0, 1$) in this particular case.

5 Quadratic Normal Alignment

In this section we will consider the problem of aligning the normals of an m -surface to vector valued data. For instance, a common problem in image

analysis is to align curves to the edges in an image, I , defined as the locations with high image gradients. This is a fundamental problem within many applications such as e.g., image segmentation. In [14] Kimmel and Bruckstein proposed to use the following functional

$$E(\Gamma) = - \int_{\Gamma} |\mathbf{n} \cdot \mathbf{v}| d\sigma, \quad (35)$$

where $\mathbf{v}$ is a differentiable vector field, e.g., the image gradient $\mathbf{v} = \nabla I$. Minimizing this functional will place Γ at locations with large values of $|\mathbf{v}|$ and simultaneously align the normal to the vector field. The reason for using the absolute value function is to make the alignment contrast-independent, that is, it does not matter if there are dark objects on a bright background, or bright objects against a dark background.

The alignment functional (35) has been analyzed in some detail by the authors in [17]. Among other things it was found that, (a) there exists admissible curves Γ where the functional is *not* Gâteaux differentiable, and (b), even when the differential $dE(\Gamma)$ does exist at Γ , then it is *not* necessarily representable by a gradient $\nabla E(\Gamma)$, in the sense of Section 2.2.

Let us elaborate these two assertions a little further. The problem of non-differentiability (a) occurs in the case when a part of the curve Γ is parallel to the vector field $\mathbf{v}$, that is, an integral curve of $\mathbf{v}$. However, such curves are far from optimal, so the problem is not a great one; one can still use differential calculus, in particular descent PDEs, to minimize (35). The problem (b), with the missing gradient interpretation of the differential, has to do with the structure of $dE(\Gamma)$ at points on the curve where the flux $\mathbf{v} \cdot \mathbf{n}$ changes its sign. In [17] it was shown that if E is differentiable at Γ , then

$$dE(\Gamma)v = \int_{\Gamma} \text{sign}(\mathbf{v} \cdot \mathbf{n})(\nabla \cdot \mathbf{v})v d\sigma + 2 \sum_{\mathbf{p}} \text{ind}(\mathbf{p})(\mathbf{v}(\mathbf{p}) \cdot \mathbf{t}(\mathbf{p}))v(\mathbf{p}),$$

for all normal velocities v on Γ . Here the sum is taken over points $\mathbf{p} \in \Gamma$ where the flux is zero. The index $\text{ind}(\mathbf{p})$ is defined in the following manner: Suppose Γ is positively oriented⁴, then the value of the index is +1 if the flux goes from negative to positive at $\mathbf{p}$, when passing along Γ in the direction of the orientation, and -1 if the flux goes from positive to negative. The vector $\mathbf{t}$ is the unit tangent vector pointing in the direction of Γ 's orientation. Thus, at each point where the flux vanishes a term of the form $\text{ind}(\mathbf{p})(\mathbf{v} \cdot \mathbf{t})\delta_{\mathbf{p}}$ turns up in the differential. Dirac distributions of this form cannot be expressed in terms of the scalar product $\langle \cdot, \cdot \rangle_{\Gamma}$ defined in Section 2.2, so if the second term in $dE(\Gamma)$ is non-zero, then there is no gradient $\nabla E(\Gamma)$ at Γ .

If there is no gradient at Γ , then there is no well-defined gradient descent either. However, there exists many descent directions in the sense of Section 3. For example, the normal velocity

⁴Going along Γ you should have its interior to your left.

$$v = -\text{sign}_\varepsilon(\mathbf{v} \cdot \mathbf{n})(\nabla \cdot \mathbf{v}) ,$$

where sign_ε is any regularization of the sign-function satisfying $\text{sign}_\varepsilon(0) = 0$ and $\text{sign}_\varepsilon(x) = \text{sign}(x)$ for $|x| > \varepsilon$, is a good descent direction for (35). We refer the reader to [17] for details.

One way to overcome the problem with a non-differentiable functional is to use a quadratic term instead. The gradient can be derived from the formulas in Section 2.2. Let us first consider a general case of “quadratic” normal alignment

$$E(\Gamma) = \int_\Gamma f(\mathbf{n} \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{w}) d\sigma = \int_\Gamma g(\mathbf{x}, \mathbf{n}) d\sigma , \quad (36)$$

where $f = f(\mathbf{x})$ is a real valued function, $f : \mathbf{R}^{m+1} \rightarrow \mathbf{R}$, and $\mathbf{v} = \mathbf{v}(\mathbf{x})$ and $\mathbf{w} = \mathbf{w}(\mathbf{x})$ are vector fields, $\mathbf{v}, \mathbf{w} : \mathbf{R}^{m+1} \rightarrow \mathbf{R}^{m+1}$.

From Theorem 1 we know that the gradient of (36) is given by $\nabla \cdot [\nabla_{S^m} g + g\mathbf{n}]$. With $g(\mathbf{x}, \mathbf{n})$ as in (36) we have

$$\begin{aligned} \nabla_{S^m} g + g\mathbf{n} &= f[\mathbf{v}(\mathbf{n} \cdot \mathbf{w}) + (\mathbf{n} \cdot \mathbf{v})\mathbf{w} - \mathbf{n}(\mathbf{n} \cdot (\mathbf{v}(\mathbf{n} \cdot \mathbf{w}) + (\mathbf{n} \cdot \mathbf{v})\mathbf{w}))] \\ &\quad + f(\mathbf{n} \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{w})\mathbf{n} \\ &= f[\mathbf{v}(\mathbf{n} \cdot \mathbf{w}) + (\mathbf{n} \cdot \mathbf{v})\mathbf{w} - (\mathbf{n} \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{w})\mathbf{n}] , \end{aligned} \quad (37)$$

and the gradient is

$$\begin{aligned} \nabla E &= \nabla \cdot [\nabla_{S^m} g + g\mathbf{n}] = \nabla \cdot [f(\mathbf{v}(\mathbf{n} \cdot \mathbf{w}) + (\mathbf{n} \cdot \mathbf{v})\mathbf{w} - (\mathbf{n} \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{w})\mathbf{n})] \\ &= (\nabla f \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{w}) + (\mathbf{n} \cdot \mathbf{v})(\nabla f \cdot \mathbf{w}) - (\nabla f \cdot \mathbf{n})(\mathbf{n} \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{w}) \\ &\quad + f \nabla \cdot [\mathbf{v}(\mathbf{n} \cdot \mathbf{w}) + (\mathbf{n} \cdot \mathbf{v})\mathbf{w} - (\mathbf{n} \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{w})\mathbf{n}] \\ &= (\nabla f \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{w}) + (\mathbf{n} \cdot \mathbf{v})(\nabla f \cdot \mathbf{w}) - (\nabla f \cdot \mathbf{n})(\mathbf{n} \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{w}) \\ &\quad + f[(\nabla \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{w}) + (\mathbf{n} \cdot \mathbf{v})(\nabla \cdot \mathbf{w}) - \kappa(\mathbf{n} \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{w}) \\ &\quad + \mathbf{v} \cdot \nabla(\mathbf{n} \cdot \mathbf{w}) + \mathbf{w} \cdot \nabla(\mathbf{n} \cdot \mathbf{v}) - \mathbf{n} \cdot (\nabla(\mathbf{n} \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{w}) \\ &\quad + (\mathbf{n} \cdot \mathbf{v})\nabla(\mathbf{n} \cdot \mathbf{w}))] \end{aligned}$$

where as before $\kappa = \nabla \cdot \mathbf{n}$.

A special case of this is the *quadratic* normal alignment functional

$$E(\Gamma) = -\frac{1}{2} \int_\Gamma (\mathbf{n} \cdot \mathbf{v})^2 d\sigma . \quad (38)$$

The corresponding gradient is simply

$$\nabla E = -(\nabla \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{v}) + \kappa(\mathbf{n} \cdot \mathbf{v})^2/2 - \mathbf{v} \cdot \nabla(\mathbf{n} \cdot \mathbf{v}) + (\mathbf{n} \cdot \nabla(\mathbf{n} \cdot \mathbf{v}))(\mathbf{n} \cdot \mathbf{v}) ,$$

which means that the gradient evolution for minimizing (38) is

$$\frac{\partial \phi}{\partial t} = [-(\nabla \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{v}) + \kappa(\mathbf{n} \cdot \mathbf{v})^2/2 - \mathbf{v} \cdot \nabla(\mathbf{n} \cdot \mathbf{v}) + (\mathbf{n} \cdot \nabla(\mathbf{n} \cdot \mathbf{v}))(\mathbf{n} \cdot \mathbf{v})] |\nabla \phi| . \quad (39)$$

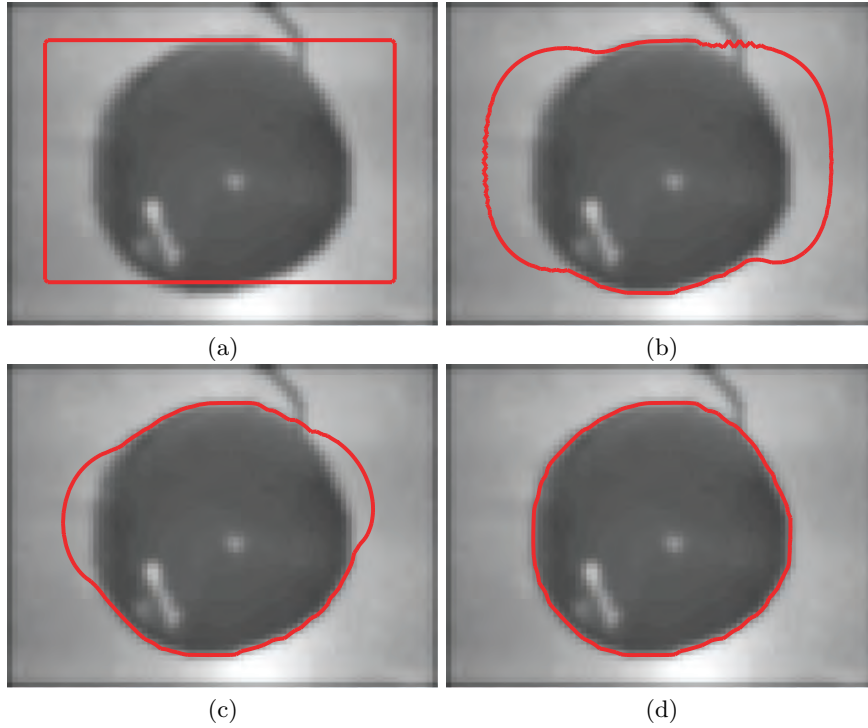


Fig. 3. An example illustrating curve alignment to image edges using the quadratic functional (38). The curve is attracted to regions with high intensity gradient and aligned so that the curve normal is parallel to the image gradient. (a) initial curve, (b) after 500 iterations, (c) after 1500 iterations, (d) after 2500 iterations.

Example. Figure 3 shows an example of using the quadratic alignment functional for aligning a curve to image edges with the evolution equation (39) where a regularization term $\alpha|I|$, $\alpha > 0$, was added to the functional to smooth the curve. In this case the vector field is the image gradient $\mathbf{v} = \nabla I$. This evolution gives very strong alignment to edges since the functional has quadratic dependence on the length of the gradient $|\nabla I|$. This means that strong edges are preferred.

6 Computing Gâteaux Derivatives using Shape Gradients

It has been suggested by [1] that the Gâteaux derivative of functionals, such as (19), can be computed in a simpler and more natural fashion by using the concept of *shape derivatives* [6] instead of standard procedures from the

calculus of variations. For the readers who want to compare the level set computations, used in this paper, with the ones using the shape derivative tool, we present the derivation, given in [1], of the Gâteaux derivative of the region-dependent functional (19),

$$E(\Omega) = \int_{\Omega} f(\mathbf{x}) \, d\mathbf{x} . \quad (40)$$

(It is convenient for our purpose to write $E(\Omega)$ instead of $E(\Gamma)$.)

To differentiate the functional (40) with respect to the shape Ω , we consider a differentiable “deformation” $t \mapsto \Omega(t)$ of Ω , defined for times t in a neighborhood of 0, and with $\Omega(0) = \Omega$, and compute

$$\left. \frac{d}{dt} E(\Omega(t)) \right|_{t=0} . \quad (41)$$

The deformed domain $\Omega(t)$ is parametrized by points in Ω in the sense that there exists a bijective mapping $T(t, \cdot) : \Omega \rightarrow \Omega(t) \subset \mathbf{R}^{m+1}$, which we shall assume differentiable, for simplicity. Moreover, at time $t = 0$,

$$T(0, \mathbf{x}) = \mathbf{x} \quad \text{for all } \mathbf{x} \in \Omega ,$$

which means that $T(0, \cdot) = \text{Id}$, the identity map on Ω .

For each $\mathbf{x} \in \Omega$ fixed, $t \mapsto T(t, \mathbf{x})$ defines a differentiable curve, whose velocity is denoted

$$\mathbf{V}(t, \mathbf{x}) = \dot{T}(t, \mathbf{x}) \quad (\mathbf{x} \in \Omega),$$

where $\dot{\cdot} = d/dt$. In particular, $\mathbf{V}(0, \cdot) : \Omega \rightarrow \mathbf{R}^{m+1}$ is a vector field which describes the “flow” of the points in Ω at time $t = 0$.

With these notions defined we are now ready to compute the derivative (41). First, we use the change of variables formula for multiple integrals to “freeze” the domain of integration in our problem:

$$E(\Omega(t)) = \int_{\Omega} f(T(t, \mathbf{x})) \det[DT(t, \mathbf{x})] \, d\mathbf{x},$$

where $DT(t, \mathbf{x})$ is the Jacobian matrix evaluated at $\mathbf{x} \in \Omega$. We may now differentiate under the integral sign.

$$\begin{aligned} \left. \frac{d}{dt} E(\Omega(t)) \right|_{t=0} &= \int_{\Omega} \left\{ \nabla f(T(t, \mathbf{x})) \cdot \dot{T}(t, \mathbf{x}) \det[DT(t, \mathbf{x})] + \right. \\ &\quad \left. + f(T(t, \mathbf{x})) \frac{d}{dt} \det[DT(t, \mathbf{x})] \right\} d\mathbf{x} \Big|_{t=0} \\ &= \int_{\Omega} \left\{ \nabla f(\mathbf{x}) \cdot \dot{T}(0, \mathbf{x}) + f(\mathbf{x}) \text{tr}[D\dot{T}(0, \mathbf{x})] \right\} d\mathbf{x} \\ &= \int_{\Omega} \left\{ \nabla f(\mathbf{x}) \cdot \mathbf{V}(0, \mathbf{x}) + f(\mathbf{x}) \nabla \cdot \mathbf{V}(0, \mathbf{x}) \right\} d\mathbf{x} \\ &= \int_{\Omega} \nabla \cdot [f(\mathbf{x}) \mathbf{V}(0, \mathbf{x})] d\mathbf{x} \\ &= \int_{\Gamma} f(\mathbf{x}) \mathbf{V}(0, \mathbf{x}) \cdot \mathbf{n} \, d\sigma . \end{aligned}$$

In the second equality above we used that the derivative $(d/dt) \det[A(t)]|_{t=0}$, where $A(t)$ is a time-dependent square matrix with $A(0) = I$, is $\text{tr}[\dot{A}(0)]$, and in the third equality, that $\text{tr}[D\mathbf{V}] = \nabla \cdot \mathbf{V}$. Since $\mathbf{V}(0, \mathbf{x}) \cdot \mathbf{n} = v$ is precisely the normal velocity of the evolution $t \mapsto \Omega(t)$, we recover the result of Theorem 2.

7 Conclusions

This paper clarified details regarding the geometric framework for variational m -surface problems. The notion of descent directions for minimizing flows was analyzed. It was shown that sometimes there exists an optimal direction and that it is possible to compare descent directions.

Furthermore, region-based functionals and normal alignment were studied in the geometric framework and new results were shown for region quotients and quadratic alignment. The connection to shape gradients was also mentioned.

Acknowledgement. The authors would like to thank Adam Karlsson for providing the images for the example in Section 4.1.

References

1. G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson. Image segmentation using active contours: Calculus of variations or shape gradients? *SIAM J. Appl. Math.*, 63, 2003.
2. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. J. Comput. Vis.*, 1997.
3. T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10(2):266–277, 2001.
4. G. Charpiat, R. Keriven, J.-P. Pons, and O. Faugeras. Designing spatially coherent minimizing flows for variational problems based on active contours. In *International Conference on Computer Vision, ICCV 2005, Beijing, China*, pages 1403–1408. IEEE Computer Society, 2005.
5. L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, 1991.
6. M. C. Delfour and J.-P. Zolesio. *Shapes and Geometries. Analysis, Differential Calculus, and Optimization*. Advances in Design and Control. SIAM, 2001.
7. A. Dervieux and F. Thomasset. A finite element method for the simulation of Rayleigh–Taylor instability. In R. Rautman, editor, *Approximation Methods for Navier–Stokes Problems*, volume 771 of *Lecture Notes in Mathematics*, pages 145–158. Springer, Berlin, 1979.
8. M. Droske and M. Rumpf. A level set formulation for willmore flow. *Interfaces and Free Boundaries*, 6(3):361–378, 2004.
9. B. Goldlücke and M. Magnor. Weighted minimal hypersurfaces and their applications in computer vision. In *European Conference on Computer Vision*, volume 2, pages 366–378, Prague, Czech Republic, 2004.

10. L. Hörmander. *The Analysis of Linear Partial Differential Operators I*. Springer Verlag, Berlin, second edition, 1990.
11. A. Karlsson and N. C. Overgaard. Theory for variational area-based segmentation using non-quadratic penalty functions. In *International Conference on Computer Vision and Pattern Recognition, San Diego, CA*. IEEE Computer Society, June 2005.
12. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Comput. Vis.*, 1(4):321–331, 1987.
13. R. Kimmel. Geometric segmentation of 3d structures. In *International Conference on Image Processing (ICIP), Barcelona, Spain*, 2003.
14. R. Kimmel and A.M. Bruckstein. Regularized Laplacian zero crossings as optimal edge integrators. *Int. J. Comput. Vis.*, 53(3):225–243, 2003.
15. D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–685, 1989.
16. S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.
17. N.C. Overgaard and J. E. Solem. An analysis of variational alignment of curves in images. In *The 5th International Conference on Scale Space and PDE methods in Computer Vision, Scale Space 2005, Hofgeismar, Germany*. Springer, 2005.
18. N. Paragios and R. Deriche. Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision. *J. Vis. Comm. Image Represent.*, 2000.
19. J. E. Solem and N.C. Overgaard. A geometric formulation of gradient descent for variational problems with moving surfaces. In *The 5th International Conference on Scale Space and PDE methods in Computer Vision, Scale Space 2005, Hofgeismar, Germany*, pages 419–430. Springer, 2005.
20. X-C. Tai and T. Chan. A survey on multiple level set methods with applications for identifying piecewise constant functions. *Int. J. Numer. Anal. Model.*, 1(1):25–47, 2004.
21. J. A. Thorpe. *Elementary Topics in Differential Geometry*. Springer-Verlag, 1985.
22. C. Xu and J. L. Prince. Snakes, shapes and gradient vector flow. *IEEE Trans. Image Process.*, 7(3):359–369, 1998.
23. A. Yezzi and A. Mennucci. Conformal metrics and true gradient flows for curves. In *International Conference on Computer Vision, ICCV 2005, Beijing, China*, pages 913–919. IEEE Computer Society, 2005.
24. H.K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127:179–195, 1996.
25. H.K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and non-parametric shape reconstruction from unorganized points using a variational level set method. In *Computer Vision and Image Understanding*, pages 295–319, 2000.

Fast PCLSM with Newton Updating Algorithm

Xue-Cheng Tai¹ and Chang-Hui Yao²

¹ CIPR and Department of Mathematics, University of Bergen, Norway. E-mail: Tai@mi.uib.no

² CIPR and Department of Mathematics, University of Bergen, Norway. E-mail: changhui.yao@cipr.uib.no

Summary. In this work, we try to develop a fast algorithm for piecewise constant level set method (PCLSM) applied to Mumford-Shah image segmentation. Just one level set function is needed to identify arbitrary number of phases for the segmentation problem. For the Mumford-Shah image segmentation model with PCLSM, one needs to minimize a smooth energy functional under some constraints. In order to solve the minimization problem, fast Newton updating algorithm is used to solve the Euler-Lagrangian equation. Due to the special structure of the segmentation functional, the cost for the Newton updating algorithm is nearly the same as for gradient updating algorithm. However, the convergence rate is much faster with a good initial guess. Numerical experiments show the efficiency and advantages of this algorithm.

Key words: PCLSM, Level set method, image segmentation, fast algorithm, Newton method.

1 Introduction

The level set method proposed by Osher and Sethian [18] is a versatile tool for tracing interfaces separating a domain Ω into subdomains. Interfaces are treated as the zero level set of some functions. Moving the interfaces can implicitly be done by evolving the level set functions instead of moving the interfaces directly. For a recent survey on the level set methods see [22, 2, 17, 25].

In [10, 11, 12] some variants of the level set method of [18], the so-called “piecewise constant level set method (PCLSM)”, was proposed to identify arbitrary number of subdomains using just one level set function. The method can be used for different applications. In [10, 11, 12], the ideas have been used for image segmentation. In [16, 24], applications to inverse shape identification problems involving elliptic and reservoir equations are shown. In this paper,

we apply PCLSM to image segmentation. Its goal is to partition a given image into regions which contain distinct objects. Different efforts have been tried to accelerate the convergence of the algorithms. In this work, we shall try to propose a Newton method which needs nearly the same cost as steepest gradient descent method, but has a much faster convergence. Let us note that Newton-type of methods have been used for the traditional level set method [17, 19, 3] in order to get shape derivatives. In our approach, no derivatives with respect to shapes are needed.

Before we go any further, we want to mention some recent related approaches that have been used in the literature for image segmentation, [11, 9, 21, 20, 8, 7]. The so-called “Binary Level Set” method as in [11, 9, 21, 20, 8] is more related to the phase field models. The model of [7] use multilayers, instead the constant values, and multiple level set functions to represent the phases.

This paper is organized in the following way. In Section 2, we review the piecewise constant level set method. In Section 3 a faster Newton updating algorithm is proposed. Details are supplied to show that the cost for this algorithm is nearly the same as for the simple steepest gradient descent scheme. In Section 4, numerical experiments are given to show the efficiency of the proposed algorithm.

2 PCLSM for Image Segmentation

We shall first recall PCLSM of [10]. The essential idea of PCLSM is to use a piecewise constant level set function to identify the subdomains. Assume that we need to partition the domain Ω into subdomains Ω_i , $i = 1, 2, \dots, n$ and the number of subdomains is a priori known. In order to identify the subdomains, we try to identify a piecewise constant level set function ϕ such that

$$\phi = i \quad \text{in } \Omega_i, \quad i = 1, 2, \dots, n. \quad (1)$$

Thus, for any given partition $\{\Omega_i\}_{i=1}^n$ of the domain Ω , it corresponds to a unique piecewise constant level set function ϕ which takes the values $1, 2, \dots, n$. Associated with such a level set function ϕ , the characteristic functions of the subdomains are defined as

$$\psi_i = \frac{1}{\alpha_i} \prod_{j=1, j \neq i}^n (\phi - j), \quad \alpha_i = \prod_{k=1, k \neq i}^n (i - k). \quad (2)$$

If ϕ is given as in (1), we have $\psi_i(x) = 1$ for $x \in \Omega_i$, and $\psi_i(x) = 0$ elsewhere. We can use the characteristic functions to extract geometrical information for the subdomains and the interfaces between the subdomains. For example,

$$\text{Length}(\partial\Omega_i) = \int_{\Omega} |\nabla \psi_i| dx, \quad \text{Area}(\Omega_i) = \int_{\Omega} \psi_i dx. \quad (3)$$

In fact, the level set function also satisfies the relation $\phi = \sum i\psi_i$. Define

$$K(\phi) = (\phi - 1)(\phi - 2) \cdots (\phi - n) = \prod_{i=1}^n (\phi - i). \quad (4)$$

At every point in Ω , the level set function ϕ satisfies

$$K(\phi) = 0. \quad (5)$$

This level set idea has been used for Mumford-Shah image segmentation in [10]. For a given digital image $u_0 : \Omega \mapsto R$ which may be corrupted by noise and blurred, the piecewise constant Mumford-Shah segmentation model is to find curves Γ and constant values c_i to minimize:

$$\sum_i \int_{\Omega_i} |c_i - u_0|^2 dx + \beta |\Gamma|. \quad (6)$$

The curves Γ separate the domain Ω into subdomains Ω_i and $\Omega = \cup_i \Omega_i \cup \Gamma$. In Chan-Vese [4], the traditional level set idea of [18] was used to represent the curves Γ and to solve the problem (6). In [10], PCLSM was used for the Mumford-Shah model (6). Note that a function u given by:

$$u = \sum_{i=1}^n c_i \psi_i \quad (7)$$

is a piecewise constant function and $u = c_i$ in Ω_i if ϕ is as given in (1). The sum in u involves characteristic functions of polynomial functions of order $n-1$ in ϕ and the unknown coefficient c_i . Each ψ_i is expressed as a product of linear factors of the form $(\phi - j)$, with the i th factor omitted.

Based on the above observations, we propose to solve the following constrained minimization problem for segmenting an image u_0 :

$$\min_{\substack{\mathbf{c}, \phi \\ K(\phi)=0}} \left\{ F(\mathbf{c}, \phi) = \frac{1}{2} \int_{\Omega} |u - u_0|^2 dx + \beta \sum_{i=1}^n \int_{\Omega} |\nabla \psi_i| dx \right\}. \quad (8)$$

We see that large approximation errors will be regularized by the fidelity term $\frac{1}{2} \int_{\Omega} |u - u_0|^2 dx$. From (3), it is clear that the latter term as the regularization term suppress oscillation. The regularization parameter $\beta > 0$ control the effect of the latter term. If the image u_0 is a piecewise constant function and we take $\beta = 0$, then any minimizers of (8) will give a function u such that $u = u_0$ where u is related to the minimizers $\mathbf{c}$ and ϕ in (7).

In [10], the augmented Lagrangian method was used to solve the constrained minimization problem (8). The augmented Lagrangian functional for this minimization problem is defined as

$$L(\mathbf{c}, \phi, \lambda) = F(\mathbf{c}, \phi) + \int_{\Omega} \lambda K(\phi) dx + \frac{r}{2} \int_{\Omega} |K(\phi)|^2 dx, \quad (9)$$

where $\lambda \in L^2(\Omega)$ is the multiplier and $r > 0$ is a penalty parameter. For the augmented Lagrangian method, it is not necessary to choose the penalization parameter r very large.

For a fixed $\mathbf{c}$, the steepest gradient descent method in ϕ for the energy function $F(\mathbf{c}, \phi)$ with constraint gives the following Euler-Lagrangian equation for the level set function ϕ :

$$\phi_t = \beta \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - (u - u_0) \frac{\partial u}{\partial \phi} - \lambda K'(\phi) - rK(\phi)K'(\phi), \quad (10)$$

with boundary condition

$$\frac{\nabla \phi}{|\nabla \phi|} \cdot \mathbf{n} = 0.$$

Here $\mathbf{n}$ is the unit outer normal of the interface.

To find a minimizer for (8), we need to find the saddle points for L . The following Uzawa gradient algorithm was used in [10] to find a saddle point for $L(\mathbf{c}, \phi, \lambda)$.

Algorithm 1 Choose initial values for ϕ^0 and λ^0 . $k = 1, 2, \dots$, do:

1. Find $\mathbf{c}^k$ from

$$L(\mathbf{c}^k, \phi^{k-1}, \lambda^{k-1}) = \min_{\mathbf{c}} L(\mathbf{c}, \phi^{k-1}, \lambda^{k-1}). \quad (11)$$

2. Use (7) to update $u = \sum_{i=1}^n c_i^k \psi_i(\phi^{k-1})$.
3. Find ϕ^k from

$$L(\mathbf{c}^k, \phi^k, \lambda^{k-1}) = \min_{\phi} L(\mathbf{c}^k, \phi, \lambda^{k-1}). \quad (12)$$

4. Use (7) to update $u = \sum_{i=1}^n c_i^k \psi_i(\phi^k)$.
5. Update the Lagrange-multiplier by

$$\lambda^k = \lambda^{k-1} + rK(\phi^k). \quad (13)$$

This algorithm has a linear convergence and its convergence has been analyzed by Kunisch and Tai in [13] under a slightly different context. The algorithm has also been used by Chan and Tai in [5, 6] for elliptic inverse problems.

The minimizer $\mathbf{c}^k$ for (11) can be obtained by solving a small $n \times n$ linear algebraic system. The minimizer for (12) is normally solved by gradient descent method, i.e.,

$$\phi^{new} = \phi^{old} - \Delta t \frac{\partial L}{\partial \phi}(\mathbf{c}^k, \phi^{old}, \lambda^{k-1}). \quad (14)$$

The step size Δt is chosen by a trial and error approach and it is fixed during the whole iterative procedure. It is not necessary to solve the minimization problem (12) exactly. Gradient iteration (14) is terminated when

$$\left\| \frac{\partial L}{\partial \phi}(\mathbf{c}^k, \phi^{new}, \lambda^{k-1}) \right\|_{L^2} \leq \frac{1}{10} \left\| \frac{\partial L}{\partial \phi}(\mathbf{c}^k, \phi^{k-1}, \lambda^{k-1}) \right\|_{L^2} \quad (15)$$

is reached or else after a fixed number of iterations. To compute $\frac{dL}{d\phi}$, it is easy to see that

$$\frac{\partial L}{\partial \phi} = (u - u_0) \frac{\partial u}{\partial \phi} - \beta \sum_{i=1}^n \nabla \cdot \left(\frac{\nabla \psi_i}{|\nabla \psi_i|} \right) \psi'(\phi) + \lambda K'(\phi) + r K(\phi) K'(\phi). \quad (16)$$

It is easy to get $\partial u / \partial \phi$, $\psi'(\phi)$ and $K'(\phi)$ from (7), (2) and (4).

3 Newton Updating

Different approaches have been used to accelerate the convergence of PCLSM. Motivated by [8], the MBO projection of [15] has been applied in [23] to deal with the constraint $K(\phi) = 0$. In [23, 24, 14], some kind of “soft” MBO projection was used. In this work, we try to use a Newton method to deal with the constraint.

Given $c^k, \phi^{k-1}, \lambda^{k-1}$, the following Newton method can be used to update ϕ and λ to get ϕ^k and λ^k , c.f [1]:

$$\begin{pmatrix} \frac{\partial^2 L}{\partial \phi^2} & \frac{\partial^2 L}{\partial \phi \partial \lambda} \\ \frac{\partial^2 L}{\partial \phi \partial \lambda} & 0 \end{pmatrix} \begin{pmatrix} \phi^k - \phi^{k-1} \\ \lambda^k - \lambda^{k-1} \end{pmatrix} = - \begin{pmatrix} \frac{\partial L}{\partial \phi} \\ \frac{\partial L}{\partial \lambda} \end{pmatrix}. \quad (17)$$

In order to solve the above system, we need to invert a huge linear algebraic system due to the regularization term in (8). In many practical applications, it is often useful to replace the Hessian matrix by some approximate Hessian matrix. Our numerical experiments indicate that the following approach is rather efficient. In order to describe the approach, we define

$$Q(\mathbf{c}, \phi, \lambda) = \frac{1}{2} \int_{\Omega} |u(\mathbf{c}, \phi) - u_0|^2 dx + \int_{\Omega} \lambda K(\phi) dx + \frac{r}{2} \int_{\Omega} |K(\phi)|^2 dx. \quad (18)$$

Thus the Hessian matrix of Q is a good approximation for the Hessian matrix of L using the fact that β is normally very small. The new algorithm using Newton method is given in the following:

(Algorithm 2) Choose initial values ϕ^0, λ^0 . For $k = 1, 2, \dots$, do:

1. Find $\mathbf{c}^k$ from

$$L(\mathbf{c}^k, \phi^{k-1}, \lambda^{k-1}) = \min_{\mathbf{c}} L(\mathbf{c}, \phi^{k-1}, \lambda^{k-1}). \quad (19)$$

2. Update $u = \sum_{j=1}^n c_j^k \psi_j(\phi^{k-1})$.
3. Find ϕ^k, λ^k from

$$\begin{pmatrix} \frac{\partial^2 Q}{\partial \phi^2} & \frac{\partial^2 Q}{\partial \phi \partial \lambda} \\ \frac{\partial^2 Q}{\partial \phi \partial \lambda} & 0 \end{pmatrix} \begin{pmatrix} \phi^k - \phi^{k-1} \\ \lambda^k - \lambda^{k-1} \end{pmatrix} = - \begin{pmatrix} \frac{\partial Q}{\partial \phi} \\ \frac{\partial Q}{\partial \lambda} \end{pmatrix}. \quad (20)$$

4. Update $u = \sum_{j=1}^n c_j^k \psi_j(\phi^k)$.

In order to solve (20), we need to invert the approximate Hessian matrix

$$\tilde{H} = \begin{pmatrix} \frac{\partial^2 Q}{\partial \phi^2} & \frac{\partial^2 Q}{\partial \phi \partial \lambda} \\ \frac{\partial^2 Q}{\partial \phi \partial \lambda} & 0 \end{pmatrix} \Big|_{(\mathbf{c}^k, \phi^{k-1}, \lambda^{k-1})}.$$

It is easy to see that $\partial L / \partial \lambda = K(\phi^{k-1})$ and $\partial L / \partial \phi$ can be obtained from (16). Using the chain rule, it is true that

$$\begin{aligned} \frac{\partial^2 Q}{\partial \phi^2} &= \left(\frac{\partial u}{\partial \phi} \right)^2 + (u - u_0) \frac{\partial^2 u}{\partial \phi^2} + \lambda K''(\phi) + r((K')^2 + K K''), \\ \frac{\partial^2 Q}{\partial \phi \partial \lambda} &= \frac{\partial^2 Q}{\partial \lambda \partial \phi} = K'(\phi). \end{aligned} \quad (21)$$

To solve this algebraic system, it is equivalent to solve a 2×2 system at each grid point. Thus, the cost for Algorithm 2 is nearly the same as for Algorithm 1 at each iteration. The solving of (19) is the same as in [10]. For clarity, we briefly outline it here. As u is linear with respect to the c_i values, we see that Q is quadratic with respect to c_i . Thus the minimization problem (19) can be solved exactly. Note that

$$\frac{\partial Q}{\partial c_i} = \int_{\Omega} \frac{\partial Q}{\partial u} \frac{\partial u}{\partial c_i} = \int_{\Omega} (u - u_0) \psi_i dx \quad \text{for } i = 1, 2, \dots, n. \quad (22)$$

Therefore, the minimizer of (19) satisfies a linear system of equations $A\mathbf{c}^k = b$:

$$\sum_{j=1}^n \int_{\Omega} (\psi_j \psi_i) c_j^k dx = \int_{\Omega} u_0 \psi_i dx, \quad \text{for } i = 1, 2, \dots, n. \quad (23)$$

In the above $\psi_j = \psi_j(\phi^{k-1})$, $\psi_i = \psi_i(\phi^{k-1})$ and thus, $\mathbf{c}^k = \{c_i^k\}_{i=1}^n$ depends on ϕ^{k-1} . The matrix A and vector b are assembled at each iteration and the equation (23) is solved by an exact solver.

Some remarks about the above algorithm are given in the following.

Remark 1. In order to make convergence for algorithm 2, we need relative good initial values. There are different ways to get initial values. In our simulations, we use Algorithm 1 to get them. In fact, we can take $\|K(\phi^k)\|_{L^2}$ as the convergence criterion. Therefore, we set a constant α such that $\|K(\phi^k)\|_{L^2} \leq \alpha \|K(\phi^0)\|_{L^2}$. We take the obtained ϕ^k as the initial values for algorithm 2 and put them into it. Here, we call α termination constant. For many of the test examples, the simple scaling procedure outlines in Section 4 is good enough to make Algorithm 2 convergent.

Remark 2. Generally, we take a small value for β . If the interfaces are oscillatory, we increase the value of β . When the noise is extremely large, we take larger values of β , r and smaller Δt to keep the algorithms stable.

4 Numerical Examples

In this section, we will present some numerical examples with images that have been tested on other related algorithms. We have used the following scaling procedure to get initial values for ϕ and $\mathbf{c}$.

First, we need to determine the phase number n before we start. Once the value of n is fixed, we scale u_0 to a function between 1 and n and take this as the initial values for ϕ , i.e.,

$$\phi^0(x) = 1 + \frac{u_0(x) - \min_{x \in \Omega} u_0}{\max_{x \in \Omega} u_0 - \min_{x \in \Omega} u_0} \times (n - 1). \quad (24)$$

For Algorithm 2, we also need an initial values for $\mathbf{c}$ and it is obtained by the following technique. From ϕ^0 , we define $\tilde{\phi}^0 = 1$ if $\phi^0 \leq 1.5$, $\tilde{\phi}^0 = i$ if $\phi^0 \in (i - 1/2, i + 1/2]$, $i = 2, 3, \dots, n - 1$, and $\tilde{\phi}^0 = n$ if $\phi^0 > n - 1/2$. Use this $\tilde{\phi}^0$ as ϕ^k in (23) to get $\mathbf{c}^k$ and use it as an initial values for $\mathbf{c}$. The initial values obtain by this procedure are often good enough to get convergence for Algorithm 2. If it is not, we use them as initial values for Algorithm 1. We do a fixed number of iterations and then use the obtained image of Algorithm 1 as the initial values for Algorithm 2. In the following, we shall refer to Algorithm 1 as **gradient updating** algorithm and refer to Algorithm 2 as **Newton updating** algorithm.

We consider only two-dimensional grey scale images. To complicate the segmentation process we typically expose the original image with Gaussian distributed noise and use the polluted image as observation data u_0 . To indicate the amount of noise that appears in the observation data, we report the signal-to-noise-ratio: $\text{SNR} = \frac{\text{variance of data}}{\text{variance of noise}}$. For every example, we will use the same parameter β for gradient updating algorithm and Newton updating algorithm, that is to say, the two methods shall get the same segmentation and the same minimization function from the view point of theories.

First, we use two examples to demonstrate that Newton updating algorithm is an efficient alternative to the multiphase algorithm of [26] where standard level set formulation is utilized and that of [10] where standard PCLSM was used with the augmented Lagrangian method. We begin with an image of an old newspaper where only two phases are needed. One phase represents the characters and the other phase represents the background of the newspaper. In this test, it is enough to guarantee Newton updating algorithm convergent by using a simple scaling procedure (24) to yield the initial values. Newton updating algorithm only uses 10 iterations to obtain an image that is as good as the image produced by gradient updating algorithm at 122 iterations, where $\beta = 0.01$, $r = 1 \times 10^6$, $\Delta t = 1e - 7$. Here CPU time is 11 seconds and 76 seconds, respectively. The segmentation has been done on the whole newspaper. In order to show the results clearly, we have just plotted a small portion of the images. The results achieved with Newton updating algorithm and gradient updating algorithm are shown in the left(bottom) of Figure 1 and the right(bottom) of Figure 1 respectively. The image obtained

by the Newton updating algorithm looks the same as the one obtained by gradient updating algorithm.



Fig. 1. Segmented images by Newton updating algorithm and gradient updating algorithm. The left(top) is an old real newspaper scaled as the initial values of ϕ . The right(top) is a small partition of the convergent $\phi = 1 \vee 2$, it is a piecewise constant function. The left (bottom) is segmented image using Newton updating algorithm at 12 iterations. The right (bottom) is the segmented image using gradient updating algorithm at 122 iterations.

The next example is a 2-phase segmentation on a real car plate image. The purpose with this test is to compare the performance of different algorithms that have been used in the literature. Like in [10], we challenge the segmentation techniques by adding a large amount Gaussian distributed noise to the real image and use the polluted image in the middle(top) of Figure 2 as the observation data. We shall compare Newton updating algorithm with gradient updating algorithm. As the noise is large, the simple scaling procedure is not good enough to get convergent for Newton updating algorithm. Thus, we use gradient updating algorithm to get the initial values $\beta = 0.75, r = 2 \times 10^5, \Delta t = 7e - 9$, and input the obtained image at 250 iterations to Newton updating algorithm, which takes CPU time 14 seconds. The value of termination constant of gradient updating algorithm is $\alpha = 0.7$. It was observed that it takes 11 Newton iterations, CPU time 0.8 second, to produce a segmentation that is as good as the segmentation taking 1338 iterations produced by gradient updating algorithm, CPU time 80 seconds. Here, we can compute the average time of every iteration for gradient updating and Newton updating, which is about CPU time 0.06 second and 0.07 second, respectively. The segmented images are displayed in the left(bottom) and right(bottom) of Figure 2. This example demonstrates the efficiency of

Newton updating algorithm clearly. In Figure 3, we give a comparison of convergence rate of gradient updating algorithm with Newton updating algorithm on the L^2 norm of minimization functional $F(c^k, \phi^k)$ and the L^2 norm of constraint function $K(\phi^k)$ over $K(\phi^0)$, which shows the efficiency of Newton updating algorithm, too. We also display the intermediate segmentations at 100,400,800,1100,1200,1338 iterations respectively in Figure 4 in order to see the segmentation procedure of gradient updating algorithm clearly.



Fig. 2. A comparison Newton updating algorithm with gradient updating algorithm. The left(top) is an original observed car plate. The middle(top) is a noisy car plate with $\text{SNR} \approx 1.7$ as observed image. The right(top) is the initial value of ϕ for Newton updating algorithm. The left(bottom) is the segmented image using Newton updating algorithm at 11 iterations. The right(bottom) is the segmented image by gradient updating algorithm at 1338 iterations.

In order to show that Newton updating algorithm can also be used to identify arbitrary number of phases, we test it on a 4-phase segmentation problem. We begin with a noisy synthetic image containing 3 objects (and a background) as shown in the left(top) of Figure 5. This is the same image as in [10, 26]. We take $\beta = 0.075, r = 2 \times 10^2, \Delta t = 5e - 6$ and use 145 iterations with gradient updating algorithm to produce an initial image for Newton updating algorithm, see the left(top) of Figure 5. Here, the value of termination constant is $\alpha = 0.8$. A careful evaluation of our algorithm is reported below. The left(bottom) of Figure 5 is produced by 6 Newton iterations starting from the initial image given in the right(top) of Figure 5. Gradient updating algorithm needs 708 iterations to converge to a similar segmentation. In the end, ϕ approaches the predetermined constants $\phi = 1 \vee 2 \vee 3 \vee 4$. Each of these constants represents one unique phase as seen in the right(bottom) of Figure 5. Our result is in accordance with what was reported in [10, 26]. For some applications, we may not know the exact number

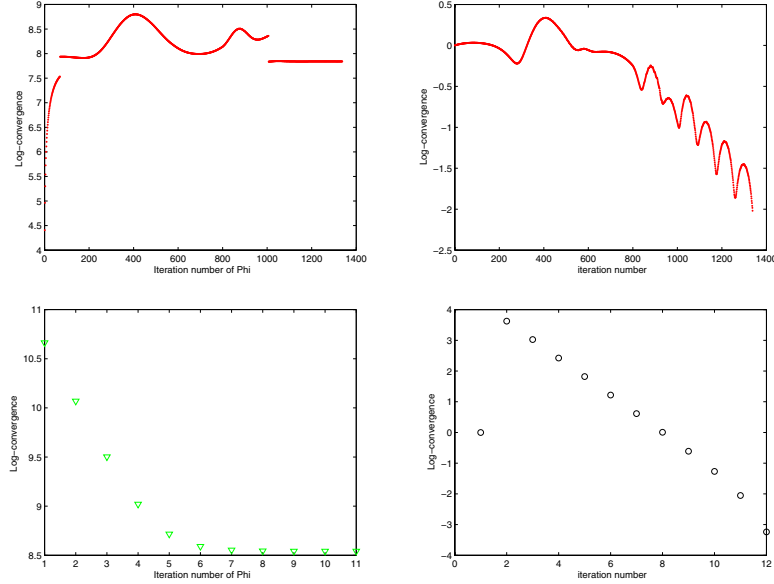


Fig. 3. The comparison of the Log_{10} -convergence of gradient updating algorithm with Newton updating algorithm. The left(top) is the Log_{10} -convergence of $\|F(c^k, \phi^k)\|_{L^2}$ with gradient updating algorithm. The right(top) is the Log_{10} -convergence for $\frac{\|K(\phi^k)\|_{L^2}}{\|K(\phi^0)\|_{L^2}}$ with gradient updating algorithm. The left(bottom) is the Log_{10} -convergence of $\|F(c^k, \phi^k)\|_{L^2}$ with Newton updating algorithm. The right(bottom) is the Log_{10} -convergence for $\frac{\|K(\phi^k)\|_{L^2}}{\|K(\phi^0)\|_{L^2}}$ with Newton updating algorithm.

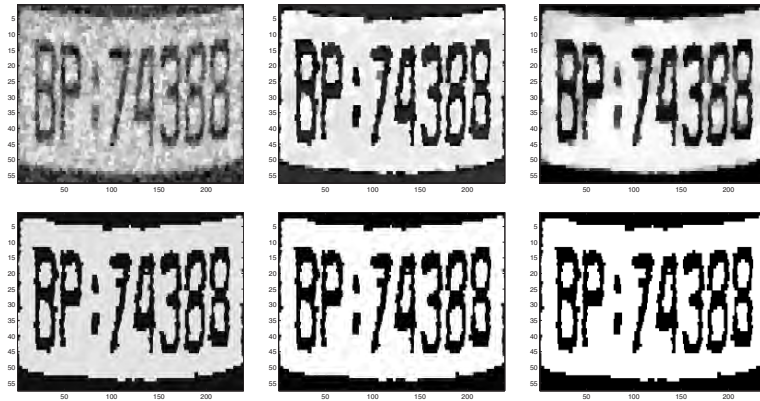


Fig. 4. The intermediate segmentations shown at 100,400,800,1100,1200,1338 iteration respectively with gradient updating algorithm.

of phases. As was demonstrated in [10], some of the phases will be empty if we take n to be bigger than 4. Some of the phases will be merged into one phase if we take n to be less than 4.

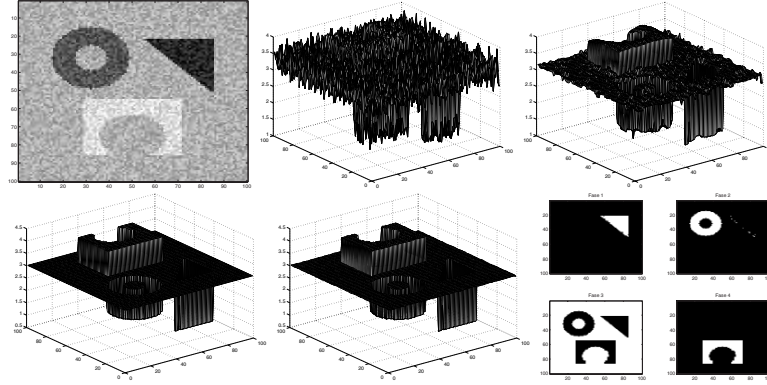


Fig. 5. A four-phase segmentation are shown to test Newton updating algorithm. The left(top) is an observed image u_0 ($\text{SNR} \approx 5.2$). The middle(top) is the initial image used for gradient updating algorithm. The right(top) is initial ϕ^0 for Newton updating algorithm produced by 147 iterations with gradient updating algorithm, CPU time is 25 seconds. The left(bottom) is the segmented image with Newton updating algorithm at 14 iterations, CPU time is 3 seconds. The middle(bottom) is the segmented image at 708 iterations with gradient updating, CPU time is 119 seconds. The right(bottom) is each segmented phase $\phi = 1 \vee 2 \vee 3 \vee 4$.

In the last example segmentation of a MR image is demonstrated. The left image in Figure 6 is available to the public at <http://www.bic.mni.mcgill.ca/brainweb/>. These realistic MRI data are used by the neuron imaging community to evaluate the performance of various image analysis methods in a setting where the truth is known. For the image used in this test the noise level is 7% and the non-uniformity intensity level of the RF-puls is 20%, see the webpage for more details concerning the noise level percent and the intensity level of the RF-puls. In Figure 7 there are three tissue classes that should be identified; phase 1: cerebrospinal fluid, phase 2: gray matter, phase 3: white matter. We take $\beta = 0.04$, $r = 0.25 \times 10^4$, $\Delta t = 5e - 6$, and use 29 iterations with gradient updating algorithm, where CPU time is 2.13 seconds, to produce an initial image for Newton updating algorithm, see the right of Figure 6. Here, the value of termination constant is $\alpha = 0.8$. Based on the initial image given in the right Figure 6, only 15 Newton iterations, where CPU time is 1.39 seconds, are needed to produce the segmented image shown in Figure 8. Compared with Figure 9 which are produced by gradient updating algorithm with 250 iterations, where CPU time is 17.63 seconds, it takes less time for Newton updating algorithm to get the same segmentation.

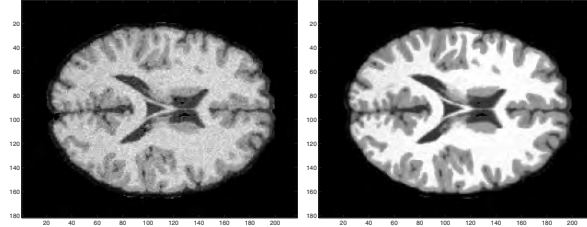


Fig. 6. The left is MRI brain image with a change in the intensity values going from left to right caused by the non-uniformity RF-puls. The right is initial image for Newton updating algorithm using 29 iterations of gradient updating algorithm.

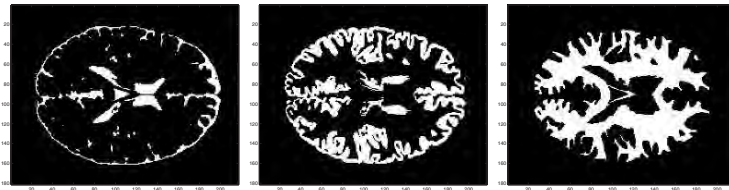


Fig. 7. The exact phases: cerebrospinal fluid, gray matter, white matter.



Fig. 8. The segmented phases with Newton updating algorithm at 15 iterations

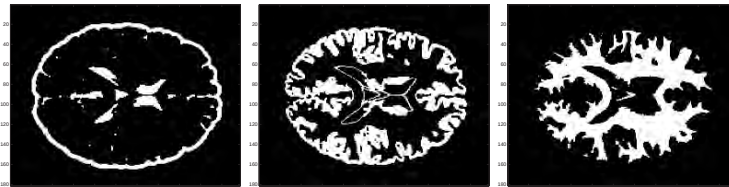


Fig. 9. The segmented are the phases with gradient updating algorithm at 250 iterations.

5 Conclusion

We have also done many other tests on Newton updating algorithm. It is confirmed that Newton updating algorithm is very fast. We can use gradient updating algorithm to produce an initial image for Newton updating algorithm. There are also many other methods that can be used to get the initial image.

Another PCLSM was proposed in [11] and it was called the Binary Level Set Method. The binary level set method extends the ideas of [9, 21] and phase field models [20]. It is clear that there is no problem to extend Newton updating algorithm to the binary level set method to accelerate the convergence.

The algorithms proposed here are able to identify arbitrary number of phase by just one level set function. Moreover, the method is easy to be extended to higher dimensional problems to segment color and video images.

References

1. D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Computer Science and Applied Mathematics. Academic Press Inc. Harcourt Brace Jovanovich Publishers, New York, 1982.
2. M. Burger and S. Osher. A survey on level set methods for inverse problems and optimal design. CAM-Report-04-02, UCLA, Applied Mathematics, 2004.
3. M. Burger, B. Hackl, and W. Ring. Incorporating topological derivatives into level set methods. *J. Comput. Phys.*, 194(1):344–362, 2004.
4. T. Chan and L. A. Vese. Active contours without edges. *IEEE Image Proc.*, 10:266–277, 2001.
5. T. F. Chan and X.-C. Tai. Level set and total variant regularization for elliptic inverse problems with discontinuous coefficients. *J. Comput. Phys.*, 193:40–66, 2003.
6. T. F. Chan and X.-C. Tai. Identification of discontinuous coefficients in elliptic problems using total variation regularization. *SIAM J. Sci. Comput.*, 25(3):881–904 (electronic), 2003.
7. J. T. Chung and L. A. Vese. Image segmentation using a multilayer level-set approach. *UCLA-CAM 03-53*, 2003.
8. S. Esedoglu and Y.-H. R. Tsai. Threshold dynamics for the piecewise constant Mumford-Shah functional. *J. Comput. Phys.*, 211(1):367–384, 2006.
9. F. Gibou and R. Fedkiw. A fast hybrid k-means level set algorithm for segmentation. *Stanford Technical Report*, 2002.
10. M. Lysaker, J. Lie and X.-C. Tai. A variant of the level set method and applications to image segmentation. CAM-Report-03-50, UCLA, Applied Mathematics, 2003.
11. M. Lysaker, J. Lie, and X.-C. Tai. A binary level set model and some applications for mumford-shah image segmentation. CAM-Report-04-31, UCLA, Applied Mathematics, 2004.

12. M. Lysaker J. Lie and X.-C. Tai. Piecewise constant level set methods and image segmentation. In Ron Kimmel, Nir Sochen, and Joachim Weickert, editors, *Scale Space and PDE Methods in Computer Vision: 5th International Conference, Scale-Space 2005*, volume 3459, pages 573–584. Springer-Verlag, Heidelberg, April 2005.
13. K. Kunisch and X.-C. Tai. Sequential and parallel splitting methods for bilinear control problems in Hilbert spaces. *SIAM J. Numer. Anal.*, 34(1):91–118, 1997.
14. H. Li and X.-C. Tai. Piecewise constant level set methods (PCLSM) for multiphase motion. Technical report, UCLA, Applied Mathematics, 2005.
15. B. Merriman, J. Bence, and S. Osher. Motion of multiple junctions: A level set approach. *J. Comput. Phys.*, 112(2):334, 1994.
16. L. K. Nielsen, X.-C. Tai, S. Aannosen, and M. E. Espedal. A binary level set model for elliptic inverse problems with discontinuous coefficients. CAM-Report-05-51, UCLA, Applied Mathematics, 2005.
17. S. Osher and R. Fedkiw. An overview and some recent results. *J. Comput. Phys.*, 169 No. 2:463–502, 2001.
18. S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.
19. J. A. Sethian. *Level set methods and fast marching methods*, volume 3 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, second edition, 1999. Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science.
20. J. Shen. Gamma-convergence approximation to piecewise constant Mumford-Shah segmentation. CAM-Report-05-16, UCLA, Applied Mathematics, 2005.
21. B. Song and T. Chan. A fast algorithm for level set based optimization. CAM-Report-02-68, UCLA, Applied Mathematics, 2002.
22. X.-C. Tai and T. F. Chan. A survey on multiple level set methods with applications for identifying piecewise constant functions. *Int. J. Numer. Anal. Model.*, 1(1):25–47, 2004.
23. X.-C. Tai, O. Christiansen, P. Lin, and I. Skjaelaaen. A remark on the MBO scheme and some piecewise constant level set methods. CAM-Report-05-24, UCLA, Applied Mathematics, 2005.
24. X.-C. Tai and H. Li. Piecewise constant level set methods (PCLSM) for elliptic inverse problems. CAM-Report-05-59, UCLA, Applied Mathematics, 2005.
25. Y.-H. R. Tsai and S. Osher. Total variation and level set methods in image science. *Acta Numer.*, 14:509–573, 2005.
26. L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *Int. J. Comput. Vis.*, 50(3):271–293, 2002.

Part IV

Fast Numerical Methods

Nonlinear Multilevel Schemes for Solving the Total Variation Image Minimization Problem

Tony F. Chan¹, Ke Chen², and Xue-Cheng Tai³

¹ Department of Mathematics, University of California, Los Angeles, CA 90095-1555, USA. E-mail: chan@ipam.ucla.edu, url: <http://www.math.ucla.edu/~chan>

² Department of Mathematical Sciences, University of Liverpool, Peach Street, Liverpool L69 7ZL, UK. E-mail: k.chen@liverpool.ac.uk, url: <http://www.liv.ac.uk/~cmchenke>

³ Department of Mathematics, University of Bergen, Bergen, Norway. E-mail: xue-cheng.tai@uib.no, url: <http://www.mi.uib.no/~tai>.

Summary. The gradient descent approach is the most widely used method in variational modeling of many image processing applications such as image restoration and segmentation. While a user is likely to be content with results obtained after a few time steps, the gradient descent approach can be quite slow in achieving convergence. Among fast iterative solvers, multilevel methods offer the potential of optimal efficiency. This paper first reviews a class of efficient numerical methods for the variational model and then presents our recent work on developing optimization multigrid methods. Advantages of the proposed algorithms over previous results are presented.

Key words: Image restoration, total variation, regularization, subspace correction, fast multilevel solvers.

AMS subject class: 68U10, 65F10, 65K10.

1 Introduction

The purpose of this paper is to address the fast solution of a variational model for image processing. To concentrate on the main ideas we consider the standard total variation (TV) based variational model which was proposed by Rudin-Osher-Fatemi (ROF) [56] and studied by various researchers [1, 73, 74, 6, 15, 44]. Other problems are equally important [19, 17, 71, 72]. We remark that improved models have recently been proposed; see [9, 14, 22, 23, 58] and references therein. Our discussion should be applicable to these new models.

The ROF TV model [56] solves the following minimisation problem

$$\min_u \int_{\Omega} \left(\alpha |\nabla u| + \frac{1}{2} (u - z)^2 \right) dx dy, \quad (1)$$

where $z = z(x, y) \in \mathbb{R}^2$ is an observed image (in practice only a discrete matrix z of $z(x, y)$ is given) that requires restoration, $u = u(x, y)$ will be the restored image, $\alpha > 0$ is a regularization parameter that is necessary for ensuring uniqueness of the inverse problem of image restoration, Ω may be taken as the unit square and $\nabla u = (u_x, u_y)$ so $|\nabla u| = \sqrt{u_x^2 + u_y^2}$. The Euler-Lagrange equation for (1) is

$$-\alpha \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + u - z = 0, \quad (2)$$

which is a nonlinear partial differential equation (PDE), also known as a curvature equation [53, 78]. One can observe that the ‘equivalence’ assumes that $|\nabla u| \neq 0$ (which is not a reasonable assumption) while problem (1) is well posed regardless $|\nabla u| \neq 0$ or not. To overcome this ‘minor’ problem, one normally solves the following equation instead of (2)

$$-\alpha \nabla \cdot \left(\frac{\nabla u}{|\nabla u|_\beta} \right) + u - z = 0, \quad (3)$$

where $|\nabla u|_\beta = \sqrt{|\nabla u|^2 + \beta}$ for some small $\beta > 0$. This equation may be viewed as the Euler-Lagrange equation for the modified problem of (1):

$$\min_u \int_\Omega \left(\alpha |\nabla u|_\beta + \frac{1}{2} (u - z)^2 \right) dx dy. \quad (4)$$

The gradient descent approach proposes to solve, instead of the elliptic PDE (3), the parabolic PDE

$$u_t = \alpha \nabla \cdot \left(\frac{\nabla u}{|\nabla u|_\beta} \right) - (u - z), \quad (5)$$

where $u = u(x, y, t)$ will converge to the solution of (3) when $t \rightarrow \infty$, with $u(x, y, 0) = z$. The advantage is that various explicit time-marching schemes may be used to solve (5) in a computationally convenient way [56, 53, 49, 72, 48]. For example, the explicit Euler scheme proceeds as follows

$$\frac{u^{k+1} - u^k}{\Delta t} = -\alpha \nabla \cdot \left(\frac{\nabla u^k}{|\nabla u^k|_\beta} \right) + u^k - z,$$

for $k \geq 0$ and $u^0 = z$. Note that if $\Delta t' = \alpha \Delta t$ can be large enough, at $k = 0$, the one-step scheme mimics the nonlinear diffusion type models [54, 43]

$$\frac{u^1 - u^0}{\Delta t'} = -\nabla \cdot \left(\frac{\nabla u^0}{|\nabla u^0|_\beta} \right).$$

As far as fast solvers are concerned, on a single level, the most robust method that we have tested for (3) is the Chan-Golub-Mulet (CGM) algorithm [26, 24] in the primal-dual pair (u, w)

$$\begin{cases} -\alpha \nabla \cdot w + u - z = 0, \\ w |\nabla u|_\beta - \nabla u = 0, \quad \|w\|_\infty \leq 1 \end{cases} \quad (6)$$

by introducing the new variable $w = \nabla u / |\nabla u|_\beta$ in a mixed formulation as in a mixed finite element method. However we shall be mainly concerned with multilevel methods in this paper for efficiently solving (1). Some numerical comparisons to this CGM algorithm are shown later on. One interesting observation of (6) is the following. Clearly eliminating w reduces it to the original PDE (3). However, if we try to eliminate u in the second equation by using $u = z + \alpha \nabla w$ from the first equation, we obtain (noting $\nabla \cdot w = \operatorname{div} w$)

$$-\nabla (\alpha \operatorname{div} w + z) + |\nabla (\alpha \operatorname{div} w + z)|_\beta w = 0$$

which reduces to the same dual formulation [14] for $\beta = 0$. Therefore, if letting $\lambda = \alpha$, the two formulations reproduce each other via their dual variables: $w = -\mathbf{p}$. (Refer to §2 below.)

2 Review of Unilevel Methods for the TV Formulation

There is a rather rich literature of related work towards efficient solution of the denoising model (1). Here we give a brief review before we turn to multilevel methods in the next section. Each method attempts to address the non-smoothness and nonlinearity in (1) in a different way.

2.1 The Dual Formulation

The primal formulation (1) may be indirectly solved via a dual formulation [14, 35]. Define the dual variable $\mathbf{p} = (p_1, p_2)$ s.t. $u = z - \lambda \operatorname{div} \mathbf{p}$. Then the dual formulation takes the form

$$\min_{\mathbf{p} \in Y} \|z - \lambda \operatorname{div} \mathbf{p}\|, \quad |\mathbf{p}_{i,j}|^2 \leq 1, \forall i, j = 1, \dots, n \quad (7)$$

where Y is the Euclidean space as specified in [14]. The above problem may be equivalently solved [14] from

$$-[\nabla(\lambda \operatorname{div} \mathbf{p} - z)]_{i,j} + |[\nabla(\lambda \operatorname{div} \mathbf{p} - z)]_{i,j}| \mathbf{p}_{i,j} = 0,$$

in which one can observe that the nonlinearity is now present in the ‘source’ term.

The dual formulation for a related problem to (1)

$$\min_u \int_\Omega \left(\alpha |\nabla u| + \frac{1}{2} (Ku - z)^2 + \frac{\beta}{2} |u|^2 \right) dx dy \quad (8)$$

is studied in [36]. Such a formulation leads to a similar dual optimization problem to (7) except that the new dual variable is bilaterally constrained.

2.2 The Modified Total Variation Method

If $|\nabla u| \neq 0$, model (1) is easy to solve. For the general case, one idea (quite different from (3)) is to modify the TV-norm [15, 40, 55] to exclude all sets where $|\nabla u| = 0$. As compensation, regularization over these sets is done with smooth norms such as with $|\nabla u|^2$. More specifically in [15], the following problem is considered:

$$\min_u \int_{\Omega} \frac{1}{2} (u - z)^2 dx dy + \alpha \left(\int_{|\nabla u| > \delta} |\nabla u| dx dy + \frac{1}{\delta} \int_{|\nabla u| \leq \delta} |\nabla u|^2 dx dy \right)$$

for a given $\delta > 0$. Although the modified problem is still non-smooth, it is formally differentiable.

Another idea of modifying the TV model is to solve the following minimization problems [5]

$$\min_u \int_{\Omega} \left(\frac{1}{2} (u - z)^2 + \frac{\alpha}{s} |\nabla u|^s \right) dx dy$$

for $1 \leq s \leq 2$ (see [23, 58] for other models of this type). Numerical solution methods for this model are proposed in [40], where the model was found to give some optimal performance with $s = 1.1$ or 1.2 . Incidentally the work of [18] on a different image problem recommends the choice of $s = 1.3$ in similarly modifying the TV norm.

2.3 The Active Set Method

This is a discrete approach [38, 13, 39] for solving the Euler-Lagrange equation of problem (8) which is a related idea to the above modified method i.e., treat inactive sets $|\nabla u| = 0$ differently from active sets $|\nabla u| > 0$. For pixels in the active sets, the problem is smooth while for others, a modified smooth problem is solved by ignoring the TV term.

2.4 The Tube Method

The discrete solution of (1) can be shown (in one dimension) to lie in a tube, bounded by two known linear splines [37]. As this solution can be interpreted as a taut string in the tube, the taut-string algorithm from statistics can solve the TV model in two dimensions [37]:

$$\left\{ \begin{array}{ll} \text{Solve } \Phi \text{ from} & \Delta \Phi = z, \quad \frac{\Phi}{n} = 0 \\ \text{Define the vector quantity} & F_z = (F_1, F_2) = \nabla \Phi \\ \text{Solve for two taut-string functions } \omega_1, \omega_2 & \text{from} \\ \min_{\omega_i} \int_{\Omega} \sqrt{1 + |\nabla \omega_i|^2} dx dy & \text{subject to the tube domain:} \\ F_1 - \alpha \leq \omega_1 \leq F_1 + \alpha, & F_2 - \alpha \leq \omega_2 \leq F_2 + \alpha. \end{array} \right.$$

Although it may appear that such a formulation is no easier than solving (1), the above method is in fact more amenable to numerical implementation than (1) because the new problem is smooth. Here $\omega = (\omega_1, \omega_2)$ acts like a dual variable but, different from [24], no β is required for (1). Moreover a fixed-point algorithm (outer-loop) is suggested [37] to solve the main nonlinear optimization step. See [59] for connections to bounded variation regularization.

2.5 The Second-Order Cone Programming Method

To derive a general method for solving (1), we note that an alternative approach is to consider

$$\min_u \int_{\Omega} |\nabla u| dx dy, \quad \text{s.t. } u + v = z, \quad \int_{\Omega} |v|^2 dx dy \leq \sigma^2,$$

where σ^2 is a variance of the noise level in z . In particular, the main TV minimization is a non-smooth problem whose discrete form may be denoted by minimizing

$$T(u_{1,1}, u_{1,2}, \dots, u_{n,n}) = \sum_{i,j=1}^n \sqrt{(u_{i,j} - u_{i+1,j})^2 + (u_{i,j} - u_{i-1,j})^2}$$

subject to the usual adjustment near the image boundaries.

The key observation made in [34] on treating the non-smooth discrete TV-term is the following: the inequality

$$\sqrt{(u_{i,j} - u_{i+1,j})^2 + (u_{i,j} - u_{i-1,j})^2} \leq t_{i,j}$$

defines a well-known second-order cone in optimization theory. The established interior point methods may be used to solve problems with such cone constraints. Therefore the proposal is to replace the minimization of T by minimizing the following equivalent merit function $\tilde{T}$

$$\begin{aligned} \tilde{T}(t_{1,1}, t_{1,2}, \dots, t_{n,n}) &= \sum_{i,j=1}^n t_{i,j}, \\ \text{s. t. } &\sqrt{(u_{i,j} - u_{i+1,j})^2 + (u_{i,j} - u_{i-1,j})^2} \leq t_{i,j} \quad \forall (i,j). \end{aligned}$$

Further the second-order cone programming method [34] is the following

$$\left\{ \begin{array}{ll} \min_{t_{1,1}, t_{1,2}, \dots, t_{n,n}} & \sum_{i,j=1}^n t_{i,j} \\ \text{s. t. } & \begin{array}{ll} u_{i,j} + v_{i,j} = z_{i,j}, & \text{for } i, j = 1, \dots, n \\ -X_{i,j} + (u_{i+1,j} - u_{i,j}) = 0, & \text{for } i = 1, \dots, n-1; j = 1, \dots, n \\ -Y_{i,j} + (u_{i,j+1} - u_{i,j}) = 0, & \text{for } i = 1, \dots, n; j = 1, \dots, n-1 \\ X_{n,k} = Y_{k,n} = 0, & \text{for } k = 1, \dots, n \\ \sqrt{X_{i,j}^2 + Y_{i,j}^2} \leq t_{i,j}, & \text{for } i, j = 1, \dots, n \\ \sqrt{v_{1,1}^2 + v_{1,2}^2 + \dots + v_{n,n}^2} \leq \sigma. \end{array} \end{array} \right.$$

Here the extra variables $X_{i,j} = (\frac{\partial u}{\partial x})_{i,j}$ and $Y_{i,j} = (\frac{\partial u}{\partial y})_{i,j}$ (and $u_{i,j}$ may be eliminated to leave $4n^2$ unknowns). See also [77]. To generate a sequence of interior points, an inner loop of iterations is introduced after putting sparsity into consideration [34]. The overall complexity is $O(N\sqrt{N})$ with $N = n^2$ for an $n \times n$ image.

2.6 The Additive Operator Splitting Method

Although we have remarked that the time-marching method is widely used (but slow), improved variants also exist. We wish to highlight the semi-implicit approach of an additive operator splitting (AOS) method which is based on classical ideas of dimensional splitting and alternating directions. The AOS method was originally proposed in [45, 46] and it was rediscovered independently later in [75] for nonlinear diffusion equations. Different properties of the AOS methods have also been studied intensively recently in [31, 33, 32, 3].

Denote the discretized version of equation (5) from a semi-implicit time-marching scheme by (in matrix vector form)

$$\frac{u^{k+1} - u^k}{\Delta t} = \sum_{\ell=1}^2 A_{\ell}(u^k) u^{k+1} \quad \text{i.e.} \quad u^{k+1} = \left(I - \Delta t \sum_{\ell=1}^2 A_{\ell}(u^k) \right)^{-1} u^k$$

where A_{ℓ} denotes the nonlinear coefficient matrix from discretization along the ℓ -coordinate direction. This is well-known. However the above inversion might not be a cheap operation if a direct (or even an iterative) method is used.

The idea of [45, 46, 75] is to make an order $O(\Delta t)$ perturbation so that the new scheme

$$u^{k+1} = \frac{1}{2} \sum_{\ell=1}^2 (I - 2\Delta t A_{\ell}(u^k))^{-1} u^k$$

is still a semi-implicit method with no essential loss of accuracy but is much easier to solve. The inversion of $(I - 2\Delta t A_{\ell}(u^k))^{-1}$ reduces to the solution of some tri-diagonal matrices over the lines parallel to the ℓ -coordinate direction, see [45, 46]. More importantly the modified scheme creates a discrete scale-space, see [75].

It should be remarked that there exist other anisotropic diffusion type models [43, 54] that are differential equations based (i.e., not minimisation based) and higher order models [47] for the same image restoration problem.

Although all above ideas might be generalized to a multilevel setting, such generalization work remains to be done. In the remainder of the paper, we shall focus on multilevel methods.

3 Review of a Class of Multigrid Methods

As we know, multigrid methods build on two well-known (i.e., old) mathematical ideas: residual (defect) correction and coarse grid approximation. The modern multigrid methods were proposed by Brandt and Hackbusch in the 1970s [28, 67]. The method was casted into a unified framework of multi-level and multidomain subspace correction in the late 1980s (see [76, 66, 62] and the references therein). See [16, 69, 52] for some recent work and refer to [28] for implementation details and example codes.

3.1 Linear Multigrid Approaches

One of the earliest attempts to solve (3) can be seen in [16, 69, 70, 71, 27, 42, 2, 60] where a linear multigrid method is used in conjunction with a linearized PDE. Essentially at the current iteration with $\bar{u}$ (starting initially from $\bar{u} = z$ with the Neumann's boundary condition), multigrid methods are used as an inner (fast) linear solver for

$$-\alpha \nabla \cdot \left(\frac{\nabla u}{|\nabla \bar{u}|_\beta} \right) + u - z = 0,$$

but the outer solver of repeating fixed point iterations may not converge very fast. There are also some other approaches using different linearization methods and solve the linearized problem by a multigrid technique.

3.2 The FAS Nonlinear Multigrid Method

One of the well known multigrid method for nonlinear problem is the FAS (Full Approximation Storage) algorithm of Brandt [10, p.346]. The original FAS algorithm for a nonlinear equation

$$N(u) = f \tag{9}$$

needs to use a sequence of nested refined meshes $\mathcal{T}_1^h, \mathcal{T}_2^h, \dots, \mathcal{T}_L^h$. Assume that $\mathcal{T}_1^h$ is the finest mesh and $\mathcal{T}_L^h$ is the coarsest mesh. For the FAS algorithm, the nonlinear equation (9) also needs to be approximated on the different meshes. Assume that equation (9) is approximated on $\mathcal{T}_k^h$ by

$$N_k(u) = f_k. \tag{10}$$

Thus, the real problem we need to solve is (10) for $k = 1$.

Consider two successive meshes on levels $k, k+1$ – a fine and a coarse level. We use R_k^{k+1} to denote the standard restriction operator between $\mathcal{T}_k^h$ and $\mathcal{T}_{k+1}^h$. Let the current approximation on level k be u_k after some smoothing steps. The task is to find a correction quantity e_{k+1} so that $u_{k+1} = R_k^{k+1}u_k + e_{k+1}$ will be the new and improved approximation on the coarser mesh on level

$k + 1$. The FAS algorithm of [10, p.346] needs to solve the following equation on level $k + 1$:

$$N_{k+1}(u_{k+1}) = \bar{f}_{k+1}, \quad (11)$$

where $\bar{f}_{k+1}$ is computed recursively through

$$\bar{f}_{k+1} = N_{k+1}(R_k^{k+1}u_k) + R_k^{k+1}(\bar{f}_k - N_k(u_k)).$$

One just needs to use a linearized smoother for equation (11) to get u_{k+1} and the correction value in fact is

$$e_{k+1} = \bar{u}_{k+1} - R_k^{k+1}u_k.$$

For our nonlinear problem (1), the solution u and the data term z are non-smooth and have discontinuities. The coarse mesh problems (10) could not approximate the problem on the finest level. Thus it may not be appropriate to use them to find the correction values over the coarser meshes unless β is sufficiently large [30, 57, 8].

3.3 Nonlinear Subspace Correction (NSSC) Methods

For linear elliptic problems, it is known that the traditional multigrid methods is the same as the subspace correction methods [76]. The subspace correction idea has been extended to nonlinear convex minimization problems in [66]. For constrained convex minimization problems, algorithms and convergence analysis are also available in [62, 64]. The essential ideas used for the nonlinear subspace correction (NSSC) methods in [66, 62, 64] can be traced back to [63, 65]; see also [25]. In the following, we shall outline the NSSC methods of [66, 62, 64] and show their differences from the FAS algorithm [10].

If we apply the NSSC of [66, 62, 64] to linear elliptic problems, it reduces to the standard multigrid method. For nonlinear problems, the essential idea of NSSC can be classified as in the following:

- The NSSC only uses the equation (10) on the finest mesh. It does not need to use the equation (10) over the coarser meshes.
- The NSSC method was formulated for finite element approximations. The functions over the coarser meshes are always regarded as a function defined on the finest mesh using the standard interpolation concept. For convex minimization problems, the corrections values need to minimize the cost functional over the finest mesh. Thus we do not need to construct cost functional over the coarser meshes.
- Nonlinear minimization problems with respect to a scalar which is the nodal value for the coarse mesh nodes to be solved over all the coarse mesh nodes. We do not need to solve these scalar nonlinear minimization problems exactly, c.f. [63]. If proper linearization methods are used for these scalar minimization problems, the cost for NSSC per iteration can be $O(N)$ where N is the degree of freedom over the finest mesh. Otherwise,

the cost is normally $O(N \log N)$ for the NSSC as all the subproblems need to be transformed to a problem over the finest mesh.

The NSSC in [66, 62, 64] was formulated for convex minimization problems. The algorithms can be extended to general nonlinear problems (1), but the convergence analysis may not be extended to (1) under general conditions. For a given reflexive Banach space V , a convex subset $K \subset V$ and a smooth convex functional $J : V \rightarrow R$, consider

$$\min_{v \in K} J(v), \quad K \subset V, \quad (12)$$

In case $K = V$, then (12) is equivalent to (9) with $N(u) = \partial J(u)$. Here ∂J is the Gateaux differential of J . Note that $N = \partial J$ is a nonlinear mapping which maps V to its dual space V^* . Assume now that we have generated a sequence of nested meshes $\mathcal{T}_k^h$. Let V_h be the finite element approximation space we shall use for (12) over the finest mesh. Then the discretized solution for (12) is the minimizer of

$$\min_{v \in V_h} J(v). \quad (13)$$

Let V_k be the finite element spaces over the meshes $\mathcal{T}_k^h$. Generally, the spaces V_k are spanned by some basis functions, i.e.

$$V_k = \text{span}(\{\phi_i^k\}_{i=1}^{n_k}) = \sum_{i=1}^{n_k} V_i^k,$$

where

$$V_i^k = \text{span}(\phi_i^k).$$

One essential idea of the NSSC is to regard V_h (i.e., V_1) as a decomposition:

$$V_h = \sum_{k=1}^L \sum_{i=1}^{n_k} V_i^k.$$

The NSSC is trying to use all the subspaces V_i^k to find the correction values. Given a current approximation u , the successive NSSC can be written as

- For $k = 1, 2, \dots, L$ and then for $i = 1, 2, \dots, n_k$:

$$\text{Find } c = \arg\min_s J(u + s\phi_i^k), \text{ and update } u \text{ as } u := u + c\phi_i^k. \quad (14)$$

- End.

As $N(u) = \partial J(u)$ is the Gateaux differential of a convex functional, the nonlinear scalar minimization problem (14) is equivalent to finding c from

$$\langle N(u + c\phi_i^k), \phi_i^k \rangle = 0.$$

Here $\langle \cdot, \cdot \rangle$ denote the duality pairing between V and V^* in the continuous setting and it is the L^2 inner product for finite element functions in the discrete setting. Thus, the following algorithm can be used for general nonlinear problems (9) and it is equivalent to the algorithm given in (14) if $N(u) = \partial J$:

- For $k = 1, 2, \dots, L$ and then for $i = 1, 2, \dots, n_k$:

Solve c from $\langle N(u + c\phi_i^k), \phi_i^k \rangle = 0$, and update u as $u := u + c\phi_i^k$. (15)

- End.

Let $g(s) = \langle N(u + s\phi_i^k), \phi_i^k \rangle$. Then (14) and (15) essentially solve

$$g(c) = 0.$$

We normally do one step of gradient descent or Newton iteration. For differentiable J functionals, it is easy to see that $g'(s) = \langle \partial^2 J(u + s\phi_i^k) \cdot \phi_i^k, \phi_i^k \rangle$. For quasilinear problems, we can also use a Picard iteration. The choice of the approximate solver for $g(c) = 0$ depends on the problem. For some problems, it is possible to solve $g(c) = 0$ in a way which only has a cost of $O(N)$ flops per iteration. For (12), we shall give some details later about how to solve (14) and (15).

It is clear that the nonlinear function g depends on i and k . For general nonlinear problems, it is very important that we do not solve (14) and (15) exactly, but replace g by some approximations depending on the problem. A proper approximation for g and a proper implementation technique can improve the numerical efficiency rather substantially. It should be observed that the functions u and ϕ_i^k are regarded as functions defined on the finest mesh. The cost functional $J(u + s\phi_i^k)$ and $N(u + s\phi_i^k)$ shall be evaluated using the values of $u + s\phi_i^k$ over the finest mesh. The duality $\langle N(u + s\phi_i^k), \phi_i^k \rangle$ is an integration involving $N(u + s\phi_i^k)$ over the support set of ϕ_i^k .

For clarity of presentation, we have chosen V_i^k to be spanned by the basis functions ϕ_i^k . For some (linear or nonlinear) problems, it might be necessary to choose the subspaces V_i^k to be spanned by a few related basis functions. We shall not go into much detail about this here.

It is preferable to avoid the use of the coarse mesh equations to get the correction values, due to several reasons. One reason is differentiability. For some non-differentiable problems, the coarse mesh problems may not approximate the fine mesh problem. The analysis of NSSC needs differentiability of the cost functional, c.f. [66, 62, 64]. However, the algorithm given in (14) can be used even for non-differentiable problems. There are also some other problems where the “simple” coarse mesh equations fail to approximate the fine mesh equation. The well-known p -Laplace equation and the equations for convection diffusion process with a non-dominating diffusion term belong to this class of problems.

3.4 FAS Based Multigrid Methods for Minimization

It is possible to use FAS algorithm for convex minimization problems at least when the minimization functional is differentiable. Several related approaches, c.f. [7, 4, 61, 11, 50], tried to design coarse grid problems by using first order

conditions to measure residuals. Specifically consider a typical setting of 2 levels: a fine level k and a coarse level $k + 1$. The ‘closeness’ of the current approximation u_k on mesh $\mathcal{T}_k^h$ to the true minimizer is measured by its first order condition (i.e., via a discrete version of (2))

$$r_k = \partial J(u_k),$$

where we assume that J must be differentiable. Thus with such a residual information available, it is proposed in [61, 50] to use the following coarse grid solver

$$\min_{u_{k+1}} J(u_{k+1}) - g_{k+1}^T u_{k+1}$$

where

$$g_{k+1} = \partial J(R_k^{k+1} u_k) - R_k^{k+1} \partial J(u_k)$$

represents the residual information projected onto the coarse grid as in a nonlinear multigrid method. This is similar to the FAS algorithm.

4 NSSC Method for Equation (1)

In order to generalize the multilevel algorithm to optimisation, we have to discuss “local relaxation” algorithms: what is a local relaxation and are local relaxations sufficient for solving (1) as a numerical method? It turns out that a local relaxation for minimisation is simply a local minimisation and local relaxations are not sufficient for solving (1) because only local non-stationary minimizers are found i.e. local relaxations can get ‘stuck’ before reaching the global minimizer. For (1), Carter [12] appears to be one of the earliest to observe such ‘stuck’ minimizers and hence would not recommend local relaxations as a standalone method. This may be seen from Figure 1 where the observed image z is denoted by $*$, the global minimizer (using $\alpha = 4$) by $\bigcirc$ and the solution from local relaxation by $\square$; clearly the local non-stationary minimizer $\square$ got stuck as noted by [12]. (Note: *a local non-stationary minimizer is not a local minimizer as the latter is also the global minimizer.*) We remark that [12] proposed hybrid relaxation methods using both the primal and dual formulations, and other ideas to avoid using the primal relaxation alone. For general convex functions, the study of block relaxation can be found in [68], where the problem of ‘stuck’ minimizers is also discussed. Other nonlinear solvers for relaxation may be found in [40, 41, 51]; in particular, sophisticated optimization methods are tested in [40].

4.1 A Piecewise Constant NSSC Algorithm for Equation (1)

Instead of attempting other relaxation methods or formulations, a different idea was considered in [20] where global minimization was achieved through utilizing multilevels. Some theoretical analysis has also been given in [21]. In

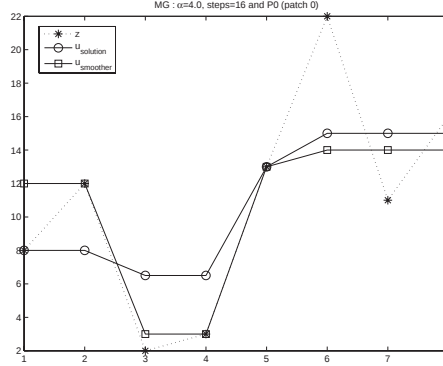


Fig. 1. Example to show that relaxations alone ($\square$) are not good methods for solution ($\circ$).

the following, we shall cast the standard approaches in [20, 21] as NSSC algorithms for piecewise constant approximation and their suggested approaches as new NSSC algorithms for (1).

Denote an observed image by $z \in \mathbb{R}^{n \times n}$ and let $n = 2^L$. The standard coarsening is used to define $L+1$ levels: $k = 1$ (finest), $2, \dots, L, L+1$ (coarsest). Take $\Omega = [0, 1] \times [0, 1]$ as the coarsest mesh, we divide each element of a coarse mesh element by connecting the four edge middle points to form four equal rectangles over the fine mesh. This gives us a nested sequence of meshes with uniform mesh sizes $h_k = 1/2^{k-1}$. The grid points for the finest mesh are $x_i = i/2^L, y_j = j/2^L$. Let $n_k = 2^{k-1}$ and $\{\tau_{i,j}^k\}_{i,j=1}^{n_k}$ be the rectangular finite elements for the mesh at level k . Then the functions $\phi_{i,j}^k$ given by

$$\phi_{i,j}^k = 1 \text{ on } \tau_{i,j}^k \text{ else } \phi_{i,j}^k = 0, \quad (16)$$

form a basis for the piecewise constant finite element space over the mesh of level k . On the finest level, the discretized minimization we shall consider is:

$$\min_u J_h(u) \quad (17)$$

where $J_h(u) = \alpha \sum_{i,j=1}^{n_k-1} \sqrt{|D_x^+ u_{i,j}|^2 + |D_y^+ u_{i,j}|^2} + \beta + \frac{1}{2} \sum_{i,j=1}^n (u_{i,j} - z_{i,j})^2$. Here u denotes a piecewise constant function defined on the finest level, $u_{i,j}$ is its value over an element $\tau_{i,j}^k$ ($k = 1$) and D_x^+, D_y^+ are the standard forward finite differences. This minimization problem is widely used for image denoising which normally works on a fixed mesh. We remark that (1) has been discretized by finite differences to give (17) so the function u may be constructed by any piecewise approximation (not restricted to piecewise constants). This is also the reason why the approximated equations on the coarser level are not appropriate to be used for the correction values.

We shall use the NSSC algorithm for solving the discretized problem (17). For this case, the algorithm given in (14) turns out to be:

Algorithm 1 (Piecewise constant NSSC algorithm)

- For $k = 1, 2, \dots, L + 1$
 - For $i, j = 1, 2, \dots, n_k$:

$$\text{Find } c = \operatorname{argmin}_s J(u + s\phi_{i,j}^k), \quad (18)$$

$$\text{and update } u \text{ as } u := u + c\phi_{i,j}^k. \quad (19)$$

- End.
- End.

For $k = 2$, one element $\tau_{i,j}^k$ is split into 4 elements on the finest level. Thus, on the finest level, $s\phi_{i,j}^k$ takes the values:

$$\begin{bmatrix} s & s \\ s & s \end{bmatrix}.$$

For an element $\tau_{i,j}^k$ on a much coarser level, the value of $s\phi_{i,j}^k$ on the finest level looks like:

$$\begin{bmatrix} s & s & \cdots & s \\ \vdots & \vdots & \vdots & \vdots \\ s & s & \cdots & s \end{bmatrix}.$$

To illustrate the setup, we show the restriction process in Figure 2 and the interpolation process in Figure 3. Here each block represents a local constant patch. On the finest level $b = 1$, each pixel is adjusted for adding the best local constant which is the same process of a local minimization (as discussed). The patch size $b \times b$ may be made variable $b_i \times b_j$ if such a set $\{(k, \ell) \mid |u_{k,\ell} - u_{i,j}| < \epsilon\}$, containing indices for a $b_i \times b_j$ block, is non-empty at the current iterate.

In [20], a one step Richardson iteration is used as an approximate solver for (18). We refer to [20] for the details about how to solve (18) in an efficient manner.

4.2 A new Piecewise Constant NSSC Algorithm with an Adaptive Subspace

In the last section, the standard multigrid subspaces for a piecewise constant finite element space are used. Due to the nature of problem (17), it was found that such a standard approach is not sufficient to achieve the global convergence. It turns out that using a new coarse mesh produced adaptively during the iterations provides a solution [20, 21]. Given u defined on the finest mesh and a threshold constant γ , we say the two adjacent grid points (x_i, y_j) and (x_{i+1}, y_j) belong to the same patch if

$$|u_{i+1,j} - u_{i,j}| \leq \gamma.$$

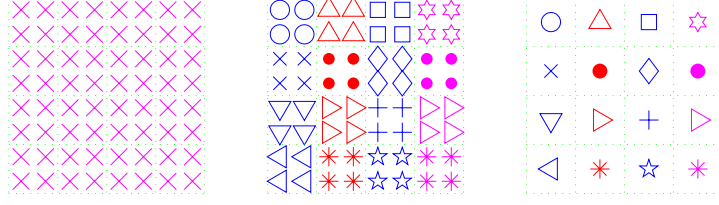


Fig. 2. Illustration of the restriction process for a piecewise constant multigrid method from the fine 8×8 grid (left) to the coarse 4×4 grid (right). Here the middle plot shows the level 2 piecewise constants and each symbol denotes a separate constant.

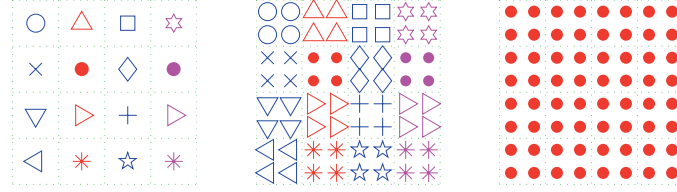


Fig. 3. Illustration of the (inherent) interpolation process for a piecewise constant multigrid method from the coarse 4×4 grid (left) to the fine 8×8 grid (right). Here the middle plot shows the interpolated level 2 piecewise constants and each symbol denotes a separate constant.

The same is used to classify two points (x_i, y_j) and (x_i, y_{j+1}) . In this way, all the grid points (x_i, y_j) can be grouped into a fixed number of patches depending on u and γ . Let $\{\Omega_i\}_{i=1}^{n_u}$ be the patches obtained using u and γ . We then define

$$\psi_i = 1 \text{ on } \Omega_i \text{ else } \psi_i = 0, \quad i = 1, 2, \dots, n_u. \quad (20)$$

We shall add the subspaces spanned by ψ_i to the decomposition for getting the correction values. With these subspaces added, the new NSSC type algorithm is:

Algorithm 2 (Adaptive piecewise constant NSSC algorithm)

- For $k = 1, 2, \dots, L + 1$
 - For $i, j = 1, 2, \dots, n_k$:

$$\text{Find } c = \operatorname{argmin}_s J(u + s\phi_{i,j}^k), \text{ and update } u \text{ as } u := u + c\phi_{i,j}^k. \quad (21)$$

- End.
- For $i = 1, 2, \dots, n_u$:

$$\text{Find } c = \operatorname{argmin}_s J(u + s\psi_i), \text{ and update } u \text{ as } u := u + c\psi_i. \quad (22)$$

- End.
- End.

The new subproblems (22) are solved using similar approximate solvers. This algorithm has been explained and analysed in detail in [21].

Returning to our earlier remark on interpretation the approximation on the finest grid, it has been proven in [13] that piecewise constant finite element functions alone cannot be used to approximate the total variation of bounded variation functions. So our classification of the above algorithm as a NSSC with piecewise constants, useful for understanding the algorithm, is not precise because the minimization (finite difference) functional (17) used in this and the last section is not the total variation of the corresponding piecewise constant function.

4.3 A Piecewise Linear Type Multilevel Algorithm

In this section, we shall explain how to use NSSC algorithm for piecewise linear finite element subspaces. For piecewise linear finite element spaces, we need to use triangular mesh over the different levels. The triangular meshes are produced from the rectangular meshes obtained in the last sections by dividing each rectangle into two triangles using the diagonal of a negative slope. Let $\phi_{i,j}^k$ be a continuous function which is a linear function over each triangular element on the k th level satisfying

$$\phi_{i,j}^k(x_i, y_j) = 1 \text{ and } \phi_{i,j}^k(x_l, y_m) = 0, l \neq i, m \neq j. \quad (23)$$

Then, $\{\phi_{i,j}^k\}_{i,j=1}^{n_k+1}$ forms a basis for the piecewise linear finite element space over level k . The number n_k is defined as before. Assume that for the given image z on the finest level with $n \times n$ pixels, the desirable image u (discrete) uniquely defines a piecewise linear function u in Ω .

If we use NSSC algorithm for the subspaces spanned by all the basis functions over all the levels as given in (23), we will get:

Algorithm 3: (Piecewise linear NSSC algorithm)

- For $k = 1, 2, \dots, L + 1$
 - For $i, j = 1, 2, \dots, n_k + 1$:

$$\text{Find } c = \operatorname{argmin}_s J(u + s\phi_{i,j}^k), \text{ and update } u \text{ as } u := u + c\phi_{i,j}^k. \quad (24)$$

- End.
- End.

The subproblems (24) are solved by approximate solvers in [29]. The only difference from the piecewise constant case is the evaluation of the values of $s\phi_{i,j}^k$. In Figure 4, the value of a basis function $\phi_{i,j}^k$ on the coarse level $k = 3$ is displayed.

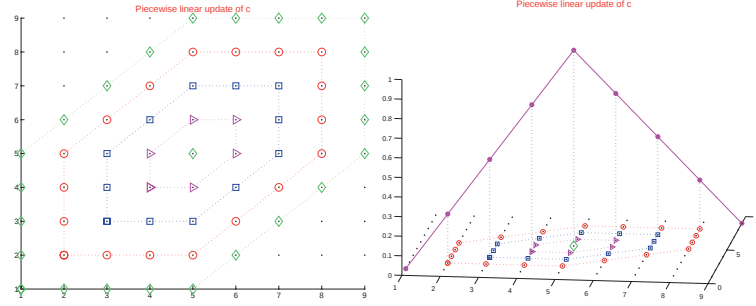


Fig. 4. A two dimensional basis function $\phi_{i,j}^k$. Note on the right plot, only the weights v_ℓ along a diagonal are shown. Here $\diamond$ defines the outer boundary of the 2D basis function, $\circ$ shows the nodes where the corresponding weights are $1/4$, $\square$ shows the nodes where the corresponding weights are $1/2$, $\triangleright$ shows the nodes where the corresponding weights are $3/4$ and the central node $\diamond$ defines the weight of 1.

Here it is important to point out that the subproblem in (24) is not expensive to solve due to the compact support of $\phi_{i,j}^k$. In fact, we can simplify the functional $J(w + s\phi_{i,j}^k)$, $s \in \mathbb{R}$, much further for efficient implementation [29]. We remark that [13] shows that piecewise linear finite element functions can be used to approximate a special variant of the standard total variation (as in (1)) of bounded variation functions. However justification for the convergence of Algorithm 3 is not yet available.

4.4 Algorithmic Complexities

For linear problems, the cost per iteration for the multigrid iteration is typically $O(N)$ flops (floating point operations), where N is the total number of degrees of freedom. For our nonlinear problems [20, 29], the cost per iteration by our Algorithm 1 is

$$2(L+1)N + (2 + 4\kappa/3)N \approx O(N \log N)$$

and by Algorithm 3 is

$$(5N + 32\kappa N)(L+1) \approx O(N \log N),$$

where we assume that $\kappa = O(1)$ steps are needed for a typical inner iteration. Here the reason why Algorithm 3 appears slightly more expensive than Algorithm 1 is that for a typical block of pixels the former only involves boundary pixels interaction while all pixels within a block in the latter interact with each other.

By way of comparison, the second order cone method [34] costs $O(N\sqrt{N})$ while most time-marching methods (including the AOS method) cost $O(\kappa N)$ where κ is the number of iterations. In the explicit Euler method, $\Delta t \approx h^2 \approx$

$1/N$. So the complexity for marching to $t = O(1)$ with $\kappa = O(N)$ will be $O(N^2)$ while the AOS method [75] is known to be 10 times faster than this. As mentioned, the cost of a fixed iteration method may not be easy to estimate as the inner solver is efficient and the outer iteration can be quite slow.

5 Numerical Experiments

To demonstrate the effectiveness of our Algorithms 2 and 3, we now present some experimental results. We remark that the above proposed algorithms have not been applied to the image minimisation problem (1) before [20, 29], although attempts on solving (4) have been made.

Effectiveness testing. We have tested the algorithms' effectiveness in solving many image denoising problems. It appears that usually a few multigrid cycles (typically 4) are sufficient to obtain an acceptable and converged result. However, readers may be more interested in comparisons with existing algorithms. Below we shall focus on this aspect. It should be remarked that some comparisons of multigrid methods with non-multilevel methods such as the fixed point iterations and time marching schemes may be found in [57]; comparison results supporting the former are not surprising in that it is faster than competitors whenever it converges.

Comparisons with an established method. There are many aspects of the discussed algorithms that could be compared with other methods. Here we choose to compare with the well-known method (perhaps the best but there are strong competitors from §2 not compared yet) of Chan-Golub-Mulet (CGM) [24]. However our task of comparing with CGM becomes somewhat easier because the CGM method 'fails' in 2 cases: (i) when the image size N becomes large (due to ill-conditioning); (ii) when $\beta \leq 10^{-32}$ (due to singularity). Here (i), not (ii), may be fixable by finding a better preconditioner (a non-trivial task) but no such work is available. In both cases, our method would converge although the local solvers take a few more iterations.

It may be of interest to show some results from parameter ranges where the CGM performs well: we take $\beta = 10^{-10}, 10^{-20}$ and 3 test examples in Figs. 5, 6 and 7. Here we mainly compare the solution's visual quality and the PSNR value which is the peak signal-to-noise ratio (PSNR) defined by (see e.g., [18])

$$\text{PSNR}(u, w) = 10 \log_{10} \frac{255^2}{\frac{1}{mn} \sum_{i,j} (u_{i,j} - w_{i,j})^2},$$

where $w_{i,j}$ and $u_{i,j}$ denote the pixel values of the restored and the original images respectively. One observes that our multilevel methods only require 3-5 cycles to obtain a comparable result.

Clearly as displayed in the vertical labels of the plots, the PSNR values of the results from our algorithms are quite close to the CGM results. Comparing CPUs is a harder task on the MATLAB platform; a more convincing test

would be to use C or Fortran in some optimal implementation. Nevertheless, our observation for the relatively small 256×256 examples is that Algorithm 1 is about 3 faster than the CGM [24] while Algorithm 3 is about as fast as [24]. This may be predicted by the complexity results shown above.

However our new algorithms are evidently more robust (without having to concern about what parameters to use) and as multilevel methods they have a scope to achieve even better performance with large images and future parallelization.

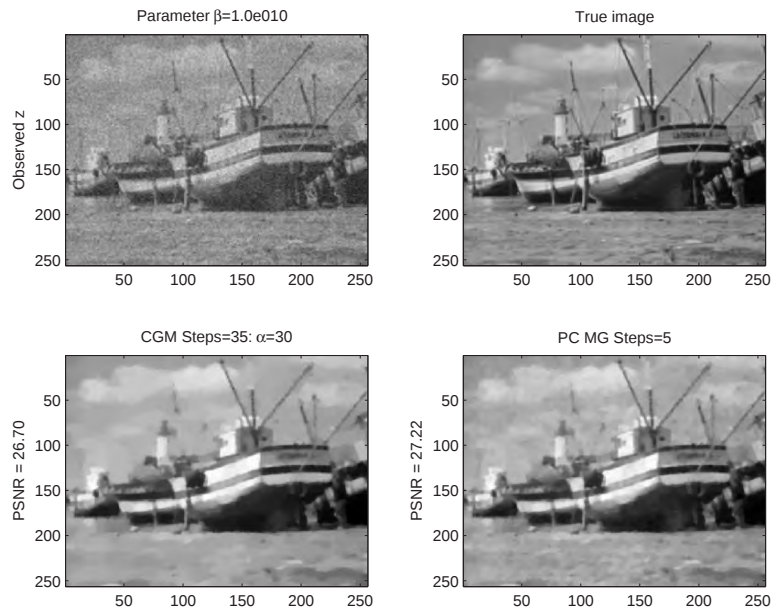


Fig. 5. Comparison of Algorithm 2 with the CGM method [24] for test example P_1 : $\alpha = 30$ and $\beta = 10^{-10}$

6 Conclusions

This paper first surveyed various solution techniques for the image denoising problem, then discussed multigrid methods for solving total variation minimization via the differential equation approach, and finally presented two related multilevel algorithms for solving total variation minimization directly. The subspace correction based algorithms differ from previous attempts for solving similar optimization problems. Numerical tests show that firstly and most importantly the new multilevel algorithms are robust and fast, and secondly they compare favorably with the well-known CGM algorithm [24], which is not a multilevel method.

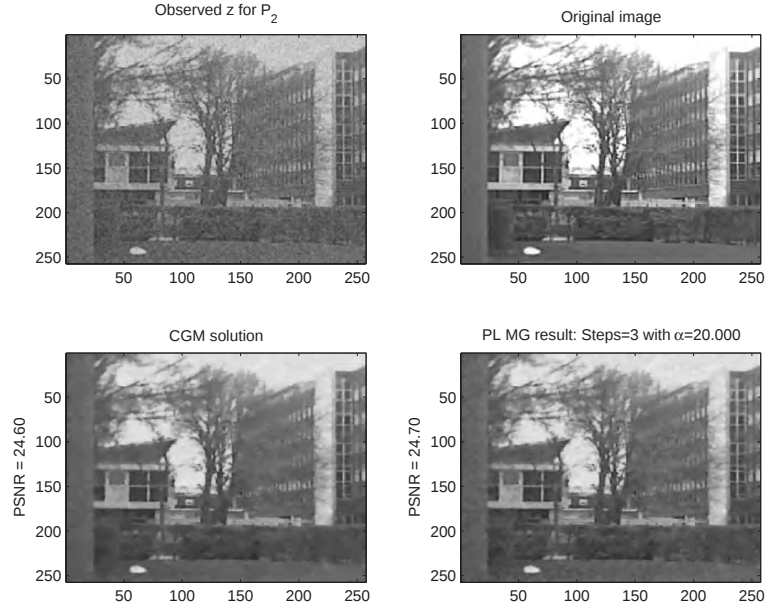


Fig. 6. Comparison of Algorithm 3 with the CGM method [24] for test example P_2 : $\alpha = 20$.

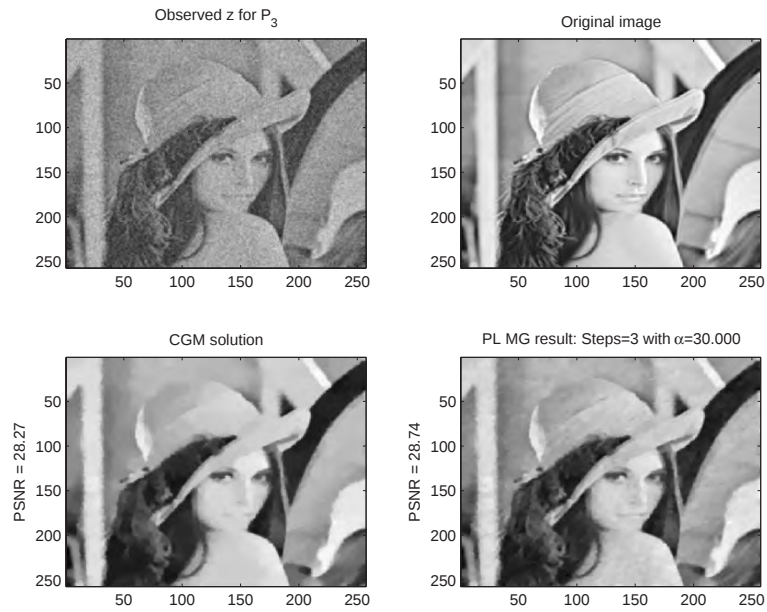


Fig. 7. Comparison of Algorithm 3 with the CGM method [24] for test example P_3 : $\alpha = 30$.

Acknowledgements

The authors thank the anonymous referees for making helpful suggestions and suggesting relevant references. This work is supported in parts by the Office of Naval Research ONR N00014-03-1-0888, the National Institutes of Health NIH U54-RR021813 and the Leverhulme Trust RF/9/RFG/2005/0482.

References

1. R. Acar and C. R. Vogel. *Analysis of total variation penalty method for ill-posed problems*, Inverse Probs., 10:1217–1229, 1994.
2. S. T. Acton. *Multigrid anisotropic diffusion*, IEEE Trans. Imag. Proc., 3 (3):280–291, 1998.
3. I. Albarreal, M. C. Calzada, J. L. Cruz, E. Fernández-Cara, J. Galo, and M. Marín. Convergence analysis and error estimates for a parallel algorithm for solving the Navier-Stokes equations. *Numer. Math.*, 93(2):201–221, 2002.
4. E. Arian and S. Ta'asan. *Multigrid one-shot methods for optimal control problems*, ICASE technical report No. 94-52, USA, 1994.
5. P. Blomgren, T. F. Chan and P. Mulet, *Extensions to Total Variation Denoising*, Proc. SPIE 97, San Diego, USA, 1997.
6. P. Blomgren, T. F. Chan, P. Mulet, L. Vese, and W. L. Wan. *Variational PDE models and methods for image processing*, in Research Notes in Mathematics, 420:43–67. Chapman & Hall/CRC, 2000.
7. A. Borzi and K. Kunisch. *A globalization strategy for the multigrid solution of elliptic optimal control problems*. *Optim. Methods Softw.* 21 (2006), no. 3, 445–459.
8. A. Bruhn, J. Weickert, T. Kohlberger and C. Schnörr, *A multigrid platform for real-time motion computation with discontinuity-preserving variational methods*, Technical Report No. 136, Department of Mathematics, Saarland University, Saarbrücken, Germany, May 2005.
9. M. Burger, S. Osher, J. Xu and G. Gilboa. *Nonlinear inverse scale space methods for image restoration*, Comm. Math. Sci., 4 (1), pp.179–212, 2006. (See also UCLA CAM report 05-34, 2005).
10. A. Brandt, *Multilevel adaptive solutions to boundary value problems*, *Math. Comp.*, pp.333–190, 1977.
11. A. Brandt, *Multigrid solvers and multilevel optimization strategies*, In J. Cong and J. R. Shinnerl, editors, *Multiscale Optimization and VLSI/CAD*, pp.1–68. Kluwer Academic (Boston), 2000.
12. J. L. Carter. *Dual method for total variation-based image restoration*, CAM report 02-13, PhD thesis, University of California at LA, USA; see <http://www.math.ucla.edu/applied/cam/index.html>.
13. E. Casas, K. Kunisch and C. Pola, *Regularization of functions of bounded variation and applications to image enhancement*, Appl. Math. Optim., 40:229–257, 1999.
14. A. Chambolle. *An algorithm for total variation minimization and applications*, J. Math. Imag. Vis., 20:89–97, 2004.
15. A. Chambolle and P.L. Lions. *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (2):167–188, 1997.

16. R. H. Chan, T. F. Chan, and W. L. Wan. *Multigrid for differential convolution problems arising from image processing*, in R. Chan, T. F. Chan, and G. H. Golub, editors, *Proc. Sci. Comput. Workshop*. Springer-Verlag, see also CAM report 97-20, UCLA, USA, 1997.
17. R. H. Chan, Q. S. Chang, and H. W. Sun., *Multigrid method for ill-conditioned symmetric Toeplitz systems*, SIAM J. Sci. Comput., 19:516–529, 1998.
18. R. H. Chan, C. W. Ho, and M. Nikolova. *Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization*, IEEE Trans. Image Proc., to appear, 2005.
19. R. H. Chan and C. K. Wong. *Sine transform based preconditioners for elliptic problems*, Numer. Linear Algebra Applic., 4:351–368, 1997.
20. T. F. Chan and K. Chen. *On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimisation*, Numer. Alg., 41:387–411, 2006.
21. T. F. Chan and K. Chen. *An optimization-based multilevel algorithm for total variation image denoising*, SIAM J. Multiscale Mod. Sim., 5(2):615–645, 2006.
22. T. F. Chan and S. Esedoglu. *Aspects of total variation regularized L^1 function approximation*, UCLA CAM report 04-07, 2004.
23. T. Chan, S. Esedoglu, F. Park and A. Yip, *Recent Developments in Total Variation Image Restoration*, in The Handbook of Math. Models in Computer Vision, eds. N. Paragios, Y. M. Chen and O. Faugeras, Springer-Verlag, pp.17-32, 2005. (See also CAM report 05-01, UCLA, USA.)
24. T. F. Chan, G. H. Golub, and P. Mulet. *A nonlinear primal dual method for total variation based image restoration*, SIAM J. Sci. Comput., 20 (6):1964–1977, 1999.
25. T. F. Chan and T. P. Mathew. *Domain decomposition algorithms*, in: Acta Numerica, ed. A. Iserles, pp.61-143, 1994.
26. T. F. Chan and P. Mulet. *Iterative methods for total variation restoration*, CAM report 96-38, UCLA, USA, 1996;
see <http://www.math.ucla.edu/applied/cam/index.html>.
27. Q. S. Chang and I. L. Chern. *Acceleration methods for total variation-based image denoising*, SIAM J. Sci. Comput., 25:982–994, 2003.
28. K. Chen. *Matrix Preconditioning Techniques and Applications*. Cambridge Monographs on Applied and Computational Mathematics (No. 19). Cambridge University Press, UK, 2005.
29. K. Chen and X.-C. Tai. *A nonlinear multigrid method for total variation minimization from image restoration*, see UCLA CAM report 05-26, USA, 2005.
30. C. Frohn-Schauf, S. Henn, and K. Witsch. *Nonlinear multigrid methods for total variation image denoising*, Comput Visual Sci., 7:199–206, 2004.
31. J. R. Galo, I. Albarreal, M. C. Calzada, J. L. Cruz, E. Fernández-Cara, and M. Marín. *Stability and convergence of a parallel fractional step method for the solution of linear parabolic problems*. AMRX Appl. Math. Res. Express, (4):117–142, 2005.
32. J. R. Galo, Isidoro I. Albarreal, M. C. Calzada, J. L. Cruz, E. Fernández-Cara, and M. Marín. *A simultaneous directions parallel algorithm for the Navier-Stokes equations*. C. R. Math. Acad. Sci. Paris, 339(3):235–240, 2004.
33. J. R. Galo, I. I. Albarreal, M. C. Calzada, J. L. Cruz, E. Fernández-Cara, and M. Marín. *Simultaneous directions parallel methods for elliptic and parabolic systems*. C. R. Math. Acad. Sci. Paris, 339(2):145–150, 2004.

34. D. Goldfarb and W. T. Yin, *Second-order cone programming methods for total variation-based image restoration*, SIAM J. Sci. Comput., 27 (2):622-645, 2005.
35. M. Hintermüller and K. Kunisch, *Total bounded variation regularization as a bilaterally constrained optimization problem*, SIAM J. Appl. Math., 64:1311-1333, 2004.
36. M. Hintermüller and G. Stadler, *An infeasible primal-dual algorithm for TV-based infconvolution-type image restoration*, Technical Report TR04-15, CAAM Dept. Rice University, USA, 2004.
37. W. Hinterberger, M. Hintermüller, K. Kunisch, M. von Oehsen and O. Scherzer, *Tube Methods for BV Regularization*, J. Math. Imaging Vis., 19:219-235, 2003.
38. K. Ito and K. Kunisch, *An active set strategy based on the augmented Lagrangian formulation for image restoration*, Math. Mod. Numer. Anal. (M2AN), 33 (1):1-21, 1999.
39. T. Kärkkäinen and K. Majava, *Nonmonotone and monotone active set methods for image restoration II. numerical results*, J. Optim. Theory Appl., 106:81-105, 2000.
40. T. Kärkkäinen, K. Majava and M. M. Mäkelä, *Comparison of formulations and solution methods for image restoration problems*, Series B Report No. B 14/2000, Department of Mathematical Information Technology, University of Jyväskylä, Finland, 2000.
41. C. T. Kelley. *Iterative Methods for Solving Linear and Nonlinear Equations*. SIAM publications, USA, 1995.
42. R. Kimmel and I. Yavneh. *An algebraic multigrid approach for image analysis*, SIAM J. Sci. Comput., 24(4):1218-1231, 2003.
43. S. H. Lee and J. K. Seo, *Noise removal with Gauss curvature driven diffusion*, IEEE Trans. Image Proc., 14 (7):904-909, 2005.
44. Y. Y. Li and F. Santosa. *A computational algorithm for minimizing total variation in image restoration*, IEEE Trans. Image Proc., 5 (6):987-995, 1996.
45. T. Lu, P. Neittaanmäki, and X.-C. Tai. *A parallel splitting up method and its application to Navier-Stokes equations*. *Appl. Math. Lett.*, 4(2):25-29, 1991.
46. T. Lu, P. Neittaanmäki, and X.-C. Tai. *A parallel splitting-up method for partial differential equations and its applications to Navier-Stokes equations*. *RAIRO Modél. Math. Anal. Numér.*, 26(6):673-708, 1992.
47. M. Lysaker, A. Lundervold and X.-C. Tai. *Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time*, IEEE Trans. Imag. Proc., 12 (12):1579-1590, 2003.
48. F. Malgouyres. *Minimizing the total variation under a general convex constraint for image restoration*, IEEE Trans. Imag. Proc., 11 (12):1450-1456, 2002.
49. A. Marquina and S. Osher. *Explicit Algorithms for a new time dependant model based on level set motion for nonlinear deblurring and noise removal*, SIAM J. Sci. Comput., 22(2):387-405, 2000.
50. S. Nash. *A multigrid approach to discretized optimisation problems*, J. Opt. Methods Softw., 14:99-116, 2000.
51. M. K. Ng, L. Q. Qi, Y. F. Yang and Y. M. Huang, *On semismooth Newton's methods for total variation minimization*, Technical Rep. 413, Dept of Math., Honk Kong Baptist Univ., China, 2005.
52. M. V. Oehsen, *Multiscale Methods for Variational Image Denoising*, Logos Verlag, Berlin, 2002.
53. S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

54. P. Perona and J. Malik, *Scale space and edge detection using anisotropic diffusion*, IEEE Trans. Pattern Anal. Mach. Intelligence, 12:629–639, 1990.
55. E. Radmoser, O. Scherzer and J. Schöberl, A cascadic algorithm for bounded variation regularization, SFB-Report No. 00-23, Johannes Kepler University of Linz, Austria, 2000.
56. L. I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, 60:259–268, 1992.
57. J. Savage and K. Chen, *An improved and accelerated nonlinear multigrid method for total-variation denoising*, Int. J. Comput. Math., 82 (8):1001–1015, 2005.
58. J. Savage and K. Chen. *On multigrids for solving a class of improved total variation based PDE models*, in this proceeding, 2006.
59. O. Scherzer. *Taut-String Algorithm and Regularization Programs with G-Norm Data Fit*, J. Math. Imaging and Vision, 23 (2):135–143, 2005
60. K. Stuben. *An introduction to algebraic multigrid*, in Appendix A of [67]. Also appeared as GMD report 70 from <http://www.gmd.de> and <http://publica.fhg.de/english/index.htm>, 2000.
61. S. Ta'asan. *Lecture note 4 of Von-Karman Institute Lectures*, Belgium, <http://www.math.cmu.edu/~shlomo/VKI-Lectures/lecture4>, 1997.
62. X.-C. Tai. Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities. *Numer. Math.*, 93:755–786, 2000.
63. X.-C. Tai and M. Espedal. *Rate of convergence of some space decomposition methods for linear and nonlinear problems*, SIAM. J. Numer. Anal., 35:1558–1570, 1998.
64. Xue-Cheng Tai, Bjorn-ove Heimsund and Jin Chao Xu. Rate of convergence for parallel subspace correction methods for nonlinear variational inequalities. In *Thirteenth international domain decomposition conference*, pages 127–138. CIMNE, Barcelona, Spain, 2002. Available online at: <http://www.mi.uib.no/7%Etai/>.
65. X.-C. Tai and P. Tseng. *Convergence rate analysis of an asynchronous space decomposition method for convex minimization*, Math. Comp., 71:1105–1135, 2001.
66. X.-C. Tai and J. C. Xu. *Global and uniform convergence of subspace correction methods for some convex optimization problems*, Math. Comp., 71:105–124, 2001.
67. U. Trottenberg, C. W. Oosterlee and A. Schuller. *Multigrid*, Academic Press, London, UK, 2000.
68. P. Tseng, *Convergence of a block coordinate descent method for nondifferentiable minimization*, J. Optim. Theory and Applica., 109 (3):475–494, 2001.
69. P. S. Vassilevski and J. G. Wade, *A comparison of multilevel methods for total variation regularization*, Elec. Trans. Numer. Anal., 6:255–270, 1997.
70. C. R. Vogel. *A multigrid method for TV-based image denoising*, in *Computation and Control IV, 20, Progress in Systems and Control Theory*, eds. K. Bowers and J. Lund, Birkhauser, 1995.
71. C. R. Vogel. *Negative results for multilevel preconditioners in image deblurring*, in *Scale-Space Theories In Computer Vision*, eds. M. Nielson et al, pp.292–304. Springer, 1999.
72. C. R. Vogel. *Computational methods for inverse problems*. SIAM publications, USA, 2002.
73. C. R. Vogel and M. E. Oman. *Iterative methods for total variation denoising*, SIAM J. Sci. Statist. Comput., 17:227–238, 1996.

74. C. R. Vogel and M. E. Oman. *Fast, robust total variation-based reconstruction of noisy, blurred images*, IEEE Trans. Image Proc., 7:813–824, 1998.
75. J. Weickert, B. M. ter Haar Romeny and M. A. Viergever, *Efficient and reliable schemes for nonlinear diffusion filtering*, IEEE Trans. Image Proc., 7:398–410, 1998.
76. J. C. Xu. *Iteration methods by space decomposition and subspace correction*, SIAM Rev., 4:581–613, 1992.
77. W. T. Yin, D. Goldfarb and S. Osher, *Image cartoon-texture decomposition and feature selection using the total variation regularized $L1$ functional*, CAM report CAM05-47, 2005, UCLA, USA.
78. A. M. Yip and F. Park. *Solution dynamics, causality, and critical behavior of the regularization parameter in total variation denoising problems*, CAM report 03-59, UCLA, USA, 2003.

Fast Implementation of Piecewise Constant Level Set Methods ^{*}

Oddvar Christiansen and Xue-Cheng Tai

Department of Mathematics, University of Bergen, Johannes Brunsgate 12, N-5007 Bergen, Norway. {oddvar, tai}@mi.uib.no

Summary. Level set methods have been proven to be efficient tools for tracing interface problems. Recently, some variants of the Osher–Sethian level set methods, which is called the Piecewise Constant Level Set Methods (PCLSM), have been proposed for some interface problems. The methods need to minimize a smooth cost functional under some special constraints. A fast algorithm for image segmentation is proposed and tested. The algorithm uses an operator splitting scheme to deal with the gradient descent equation. A special technique is used to tackle the constraint for the PCLSM. By choosing the time step and the penalization parameter properly, the cost functional is minimized and the constraint is fulfilled. Experiments for image segmentation is given. The efficiency of the algorithm and the quality of the obtained images are demonstrated.

Key words: level set method; image segmentation; total variation regularization; operator splitting

1 Introduction

A function $u(\mathbf{x})$ defined on an open and bounded domain $\Omega \in \mathbb{R}^d$ may have different properties in distinct regions of Ω . In many applications one wants to separate Ω into a union of these regions, i.e. $\Omega = \cup_{i=1}^n \Omega_i$. There are several approaches to accomplishing this segmentation, one is the successful level set method invented by Osher and Sethian [14].

In the standard level set method a distance function $\phi(\mathbf{x})$ is assigned to the function $u(\mathbf{x})$, and the interior and exterior of Ω are represented implicitly by the sign of $\phi(\mathbf{x})$. We have that the interior of Ω is represented by the points $\mathbf{x}$: $\phi(\mathbf{x}) > 0$ and the exterior of Ω is represented by the points $\mathbf{x}$: $\phi(\mathbf{x}) < 0$. The boundary is represented as the zero level set curve $\Gamma = \{\mathbf{x} \in \Omega, \phi(\mathbf{x}) = 0\}$. The advantage of this representation is that rather than evolving the curve

^{*}We acknowledge support from the Norwegian Research Council and IMS of the National University of Singapore.

itself, we evolve the distance function $\phi(\mathbf{x})$. This makes topology changes such as merging and breaking an easy task. To divide Ω into more than two domains, one needs to use multiple level set functions [25, 16, 22].

Recently, Lie, Lysaker and Tai [8] presented the piecewise constant level set method (PCLSM) as an alternative approach to the multiple level set function. This method only requires one level set function to represent multiphase segmentation. For some shape identification problems, the PCLSM needs to minimize a smooth functional under some constraints. In [8], gradient method of Uzawa type has been used to solve the saddle point problem coming from the Euler-Lagrange equation for the constrained minimization. Such a method is rather stable, but often has slow convergence. In several recent works, fast algorithms have been proposed to solve this constrained minimization problem. In [19], the MBO projection of [12] has been used to deal with the constraint. The convergence is fast, but the time step needs to be chosen carefully. In [20], a quasi-Newton approach has been tested to solve the saddle point equations. Due to the special structure of the segmentation problem, the cost per Quasi-Newton updating is nearly the same as the gradient updating, but the convergence is much faster if we have good initial guesses. In this work, we try another technique to accelerate the convergence. Due to the special structure of the constraint, we are able to design a special procedure to deal with it. By choosing the penalization parameter and the time step in a proper manner, we are able to use Newton method to enforce the constraint in a rather cost efficient way. Numerical experiments show that this technique has fast convergence and it also has better stability properties. For most of the experiments we have done, we can use the same set of parameters and the algorithm is able to converge in about 40 iterations.

The PCLSM was intended as an alternative for the traditional level set idea of [14, 3]. The ideas are also somehow related to the phase field models for phase transition [17]. They also extend the models proposed in [18, 6]. In [5], the layer between the constant levels are used to distinguish the phases. Here, we use the constant levels. Recently, similar ideas have also been proposed in [9] for some complicated inverse scattering problems.

This work is organized in the following way. In Section 2, we outline the essential ideas of the PCLSM of [8]. In order to improve the efficiency of the algorithms, we will use some operator splitting methods for our computation. A general introduction about the operator splitting methods is given in Section 3. The essential ideas for our fast algorithm is presented in Section 4. Here all the details behind the algorithm are explained. The algorithm and its essential numerical features are exposed in Section 5. We report the numerical experiments in Section 6. The tests show both the quality and the speed of the proposed algorithm.

2 Piecewise Constant Level Set Formulation

First, we give a brief outline of the PCLSM [8]. Assume that we need to find N regions $\{\Omega_i\}_{i=1}^N$ which form a partition of Ω . In order to find the regions we try to find a piecewise constant function which takes values

$$\phi = i \text{ in } \Omega_i, \quad i = 1, 2, \dots, N. \quad (1)$$

The discontinuities of ϕ give us the curves that separate the regions. Associated with ϕ we define the characteristic functions ψ_i for Ω_i as

$$\psi_i = \frac{1}{\alpha_i} \prod_{\substack{j=1 \\ j \neq i}}^N (\phi - j) \quad \text{and} \quad \alpha_i = \prod_{\substack{k=1 \\ k \neq i}}^N (i - k). \quad (2)$$

Each ψ_i is expressed as a product of linear factors of the form $\phi - j$, with the i th factor omitted. Consequently the characteristic functions ψ_i will have the property

$$\psi_i(\mathbf{x}) = \begin{cases} 1 & \text{if } x \in \Omega_i \\ 0 & \text{elsewhere} \end{cases}, \quad (3)$$

as long as (1) holds.

From the characteristic functions we can easily calculate geometric properties like length and area. The length of the boundary of Ω_i is given by the relation

$$|\partial\Omega_i| = \int_{\Omega} |\nabla \psi_i| dx, \quad (4)$$

and the area inside Ω_i is given by the relation

$$|\Omega_i| = \int_{\Omega} \psi_i dx. \quad (5)$$

By linearly combining these characteristic functions we are able to build a cartoon or a piecewise constant image,

$$u = \sum_{i=1}^n c_i \psi_i. \quad (6)$$

This is a piecewise constant function and $u = c_i$ in Ω_i if ϕ is as given in (1).

In order to guarantee that the level set function ϕ takes the values as in (1) at convergence, we introduce the constraint function

$$K(\phi) = (\phi - 1)(\phi - 2) \cdots (\phi - N) = \prod_{i=1}^N (\phi - i). \quad (7)$$

Requiring

$$K(\phi) = 0 \quad (8)$$

at convergence ensures that ϕ only takes integer values, and that each point $\mathbf{x} \in \Omega$ belongs to one and only one phase. This prevents vacuum and overlap between the different phases.

Based on the above observations we propose to solve the following Mumford-Shah functional [13] to find a segmentation of a given image u_0 :

$$\min_{\substack{\mathbf{c}, \phi \\ K(\phi)=0}} \left\{ F(\mathbf{c}, \phi) = \int_{\Omega} |u - u_0|^2 dx + \beta \sum_{i=1}^n \int_{\Omega} |\nabla \psi_i| dx \right\}. \quad (9)$$

In the above, β is a nonnegative parameter controlling the regularizing, u is a piecewise constant function depending on ϕ and $\mathbf{c}$, as in (6). The first term of (9) is a least square functional, measuring how well the piecewise constant image u approximates u_0 . The second term is a regularizer measuring the length of the edges in the image u .

A more simplified cost functional can be achieved by regularizing ϕ directly in (9). The following relation

$$c_1(N) \int_{\Omega} |\nabla \phi| dx \leq \sum_{i=1}^N \int_{\Omega} |\nabla \psi_i| dx \leq c_2(N) \int_{\Omega} |\nabla \phi| dx, \quad (10)$$

where $c_1(N)$ and $c_2(N)$ only depends on N , gives the simplified minimization problem

$$\min_{\substack{\mathbf{c}, \phi \\ K(\phi)=0}} \left\{ F(\mathbf{c}, \phi) = \int_{\Omega} |u - u_0|^2 dx + \beta \int_{\Omega} |\nabla \phi| dx \right\}. \quad (11)$$

To deal with the constraint $K(\phi) = 0$ we use a penalization method. Defining $W(\phi) = |K(\phi)|^2$ we propose the following penalization functional:

$$\min_{\mathbf{c}, \phi} \left\{ F(\mathbf{c}, \phi) = \int_{\Omega} |u - u_0|^2 dx + \beta \int_{\Omega} |\nabla \phi| dx + \frac{1}{\mu} \int_{\Omega} W(\phi) dx \right\}. \quad (12)$$

To solve this minimization problem we propose to use an operator splitting scheme combined with Newton iteration. A similar minimization problem was solved in [8], using augmented Lagrangian Method. It has also been solved in [19] using a MBO approach and in [20] using a quasi-Newton approach.

3 Operator Splitting Scheme

In this section we try to explain the operator splitting scheme in a general setting. For a given function space V and an operator (linear or nonlinear) defined in V , we often need to solve the following time dependent equation:

$$\frac{\partial \phi}{\partial t} + A(\phi) = f(t), \quad t \in [0, T], \quad \phi(0) = \hat{\phi} \in V. \quad (13)$$

If the operator A and the function f can be split in the following way:

$$A = A_1 + A_2 + \cdots + A_m, \quad f = f_1 + f_2 + \cdots + f_m, \quad (14)$$

then splitting schemes can be used to approximate the solution of (13). Normally, the operators A_i are simpler and easier to solve. The first scheme is called the parallel splitting scheme or additive operator splitting (AOS) scheme. First we choose a time step τ and set $\phi^0 = \hat{\phi}$. At each time level $t_j = j\tau$, we compute $\phi^{j+\frac{i}{2m}}$ in parallel for $i = 1, 2, \dots, m$ from:

$$\frac{\phi^{j+\frac{i}{2m}} - \phi^j}{m\tau} + A_i(\phi^{j+\frac{i}{2m}}) = f_i(t_j), \quad \text{and then set } \phi^{j+1} = \frac{1}{m} \sum_{i=1}^m \phi^{j+\frac{i}{2m}}. \quad (15)$$

This algorithm was first proposed in Lu, Neittaanmaki and Tai [10, 11]. It was discovered independently later in [23] and used in a different context for image processing [24, 2, 1].

The following sequential scheme, sometimes also called the multiplicative operator splitting (MOS) scheme can also be used to approximate the solution of (13):

$$\frac{\phi^{j+\frac{i}{m}} - \phi^{j+\frac{i-1}{m}}}{\tau} + A_i(\phi^{j+\frac{i}{m}}) = f_i(t_j), \quad i = 1, 2, \dots, m. \quad (16)$$

We are able to combine the AOS and MOS schemes in different ways, and below we present a combined scheme which will be used for our algorithms.

Split the operator A and the function f in the following way:

$$A = \underbrace{A_1 + A_2 + \cdots + A_k}_{\text{AOS part}} + \underbrace{A_{k+1} + \cdots + A_m}_{\text{MOS part}}, \quad (17)$$

$$f = \underbrace{f_1 + f_2 + \cdots + f_k}_{\text{AOS part}} + \underbrace{f_{k+1} + \cdots + f_m}_{\text{MOS part}}, \quad (18)$$

i.e. we have grouped A and f into two parts. We now can use the AOS scheme on the first k terms and then the MOS scheme on the remaining $m - k$ terms. This gives the following algorithm:

Algorithm 1 (*A General AOS-MOS scheme*)

- Use the AOS scheme on the first k terms, i.e. solve $\phi^{j+\frac{i}{2m}}$ in parallel from

$$\frac{\phi^{j+\frac{i}{2m}} - \phi^j}{k\tau} + A_i(\phi^{j+\frac{i}{2m}}) = f_i(t_j), \quad i = 1, 2, \dots, k. \quad (19)$$

- Set

$$\phi^{j+\frac{k}{m}} = \frac{1}{k} \sum_{i=1}^k \phi^{j+\frac{i}{2m}}. \quad (20)$$

- Use the MOS scheme for the remaining terms, i.e. solve $\phi^{j+\frac{i}{m}}$ sequentially from

$$\frac{\phi^{j+\frac{i}{m}} - \phi^{j+\frac{i-1}{m}}}{\tau} + A_i(\phi^{j+\frac{i}{m}}) = f_i(t_j), \quad i = k+1, k+2, \dots, m. \quad (21)$$

4 Operator Splitting and Newton Methods for Image Segmentation

In this section we show how the operator splitting idea can be used for the minimization problem (12). In order to solve (12) we need to find $\mathbf{c}$ and ϕ that satisfy

$$a) \frac{\partial F}{\partial \mathbf{c}} = 0, \quad b) \frac{\partial F}{\partial \phi} = 0. \quad (22)$$

As u is linear with respect to the c_i values, we see that F is quadratic with respect to c_i . Thus the minimization of (12) with respect to $\mathbf{c}$ can be solved exactly. We have

$$\frac{\partial F}{\partial c_i} = \int_{\Omega} (u - u_0) \psi_i \, dx, \quad \text{for } i = 1, 2, \dots, N. \quad (23)$$

Therefore, the minimizer of (12) with respect to $\mathbf{c}$ satisfies a linear system of equations $A\mathbf{c} = b$:

$$\sum_{j=1}^n \int_{\Omega} (\psi_i \psi_j) c_j \, dx = \int_{\Omega} u_0 \psi_i \, dx, \quad \text{for } i = 1, 2, \dots, N. \quad (24)$$

This can be easily solved by forming the matrix A and the vector b and solve the equation $A\mathbf{c} = b$ by an exact solver. The size of the system is very small, i.e. A is a $N \times N$ matrix.

To compute $\frac{\partial F}{\partial \phi}$ we utilize the chain rule to get

$$\frac{\partial F}{\partial \phi} = -\beta \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + (u(\phi, \mathbf{c}) - u_0) \frac{\partial u}{\partial \phi} + \frac{1}{\mu} W'(\phi). \quad (25)$$

The variational formulation also impose the following boundary condition for ϕ on Ω

$$\frac{\nabla \phi}{|\nabla \phi|} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega. \quad (26)$$

Using a steepest descent method for the minimization of (12) with respect to ϕ we get the following equation for the level set function ϕ :

$$\phi_t = \beta \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - (u(\phi, \mathbf{c}) - u_0) \frac{\partial u}{\partial \phi} - \frac{1}{\mu} W'(\phi). \quad (27)$$

This is a partial differential equation which we solve using Algorithm 1. We split the right side of (27) into $d + 1$ terms:

$$\phi_t = B_1(\phi) + B_2(\phi) + \cdots + B_d(\phi) + C(\phi), \quad (28)$$

where

$$B_i(\phi) = \beta D_i \cdot \left(\frac{D_i \phi}{|\nabla \phi|} \right) - \frac{1}{d} (u(\phi, \mathbf{c}) - u_0) \frac{\partial u}{\partial \phi}(\phi, \mathbf{c}), \quad (29)$$

and

$$C(\phi) = -\frac{1}{\mu} W'(\phi), \quad (30)$$

$D_i = \frac{\partial}{\partial x_i}$ and d is the spatial dimension. Thus, we have $m = d + 1$. Applying Algorithm 1 directly on (28), we get the following algorithm.

Algorithm 2 (*Mixed AOS-MOS scheme*). For $n = 1, 2, \dots$ until convergence.

- Use the AOS scheme on the first d terms

$$\frac{\phi^{n+\frac{i}{2m}} - \phi^n}{\tau d} = \beta D_i \cdot \left(\frac{D_i \phi^{n+\frac{i}{2m}}}{|\nabla \phi^{n+\frac{i}{2m}}|} \right) - \frac{1}{d} (u(\phi^{n+\frac{i}{2m}}, \mathbf{c}) - u_0) \frac{\partial u}{\partial \phi}(\phi^{n+\frac{i}{2m}}, \mathbf{c}) \quad (31)$$

$$i = 1, 2, \dots, d.$$

- Set

$$\phi^{n+\frac{1}{2}} = \frac{1}{d} \sum_{i=1}^d \phi^{n+\frac{i}{2m}} \quad (32)$$

- Solve ϕ^{n+1} from

$$\frac{\phi^{n+1} - \phi^{n+\frac{1}{2}}}{\tau} = -\frac{1}{\mu} W'(\phi^{n+1}). \quad (33)$$

In what follows we will show how to efficiently solve (31) and (33). The first of these equations, i.e. equation (31), is nonlinear and implicit. In order to solve it, we use the semi-implicit Picard iteration

$$\frac{\phi_i^{new} - \phi^n}{\tau d} = \beta D_i \cdot \left(\frac{D_i \phi_i^{new}}{|\nabla \phi_i^{old}|} \right) - \frac{1}{d} (u(\phi_i^{old}, \mathbf{c}) - u_0) \frac{\partial u}{\partial \phi}(\phi_i^{old}, \mathbf{c}). \quad (34)$$

For each i , we choose an initial value as ϕ_i^{old} and get a ϕ_i^{new} which is then taken to be ϕ_i^{old} and get another ϕ_i^{new} . This procedure is iterated until convergence. We then set

$$\phi^{n+\frac{1}{2}} = \frac{1}{d} \sum_{i=1}^d \phi_i^{new}. \quad (35)$$

We have chosen to use a semi-implicit scheme to improve the stability and reduce the computational time.

The reason for the dimensional splitting is that this leads to a system of equations which can be efficiently solved using direct solvers for tri-diagonal matrices. Rewrite (34) as

$$\phi_i^{new} - \tau\beta d D_i \cdot \left(\frac{D_i \phi_i^{new}}{|\nabla \phi_i^{old}|} \right) = \phi^n - \tau(u(\phi_i^{old}, \mathbf{c}) - u_0) \frac{\partial u}{\partial \phi}(\phi_i^{old}, \mathbf{c}) =: r_i, \quad (36)$$

and define the operator

$$A_i = D_i \cdot (a(\mathbf{x}) D_i), \quad (37)$$

where $a(\mathbf{x}) = \frac{1}{|\nabla \phi_i^{old}|}$. Using this we can write (34) as

$$(I - \tau\beta d A_i) \phi_i^{new} = r_i, \quad (38)$$

where I is the identity matrix. For each i , the matrix $(I - \tau\beta d A_i)$ is a tri-diagonal matrix on the mesh lines parallel to the x_i -axes. Thus the systems (38) can be solved fast using a tri-diagonal solver.

The second equation, (33), can be efficiently solved using the Newton iteration. Define:

$$G(\phi) = \phi + \frac{\tau}{\mu} W'(\phi) - \phi^{n+1/2}. \quad (39)$$

We see that (33) is the same as finding a root for G . This problem can be easily solved using the Newton iteration

$$\phi^{new} = \phi^{old} - \frac{G(\phi^{old})}{G'(\phi^{old})}. \quad (40)$$

There is however one problem to take into consideration. W' is a polynomial of degree $2N-1$, where N is the number of phases. Thus there are $2N-1$ roots. If no restriction is placed on τ and μ we can have more than one solution for the system and the Newton iteration can converge to any one of these solutions. Thus to ensure uniqueness and convergence of the Newton iteration we shall choose τ and μ so that $G' > 0$. This will ensure that G is strictly increasing and thus there is only one real root. The rest of the roots are complex. It is easy to see that

$$G'(\phi) = 1 + \frac{\tau}{\mu} |K'(\phi)|^2 + \frac{\tau}{\mu} K(\phi) K''(\phi). \quad (41)$$

Some simple calculations show that $G' > 0$ will impose the constrain in Table 1 on τ and μ . The bound depends on the number of phases N . This means that for a given μ we can easily calculate the time step τ to make $G' > 0$. In the next section we present the complete algorithm and show how to choose a proper value for the penalization parameter μ and the initial values for the constants $\mathbf{c}$.

Remark 1. We have used AOS for the B_i operators and MOS for the C operator. The reason to use AOS for B_i is to treat all the spatial variable x_i in a symmetrical way. This can avoid turning symmetrical images into non-symmetrical images.

Table 1. Upper bounds σ_0 for $\frac{\tau}{\mu}$

N	$\tau/\mu <$
2	2
3	0.71
4	0.09
...	...

5 The Algorithm

The penalization parameter μ controls the effect of the constraint $K(\phi) = 0$. When μ is very small the constraint has large impact on (12), evolving the level set function quickly towards integer values. Whereas, when μ is large the regularizing and fidelity terms are more dominant, smoothing the image under the constraint that u is close to u_0 . Our idea is that we start with a large μ ensuring that the regularizing and fidelity terms are the dominant ones. We then slowly reduce μ towards zero. This will gradually increase the impact of the constraint, ensuring that the level set function ϕ converges towards a piecewise constant function with $\phi = i$ in Ω_i .

Numerical tests we have done show that starting with μ equal to 1000 and then setting $\mu_{new} = 0.75 \cdot \mu_{old}$ for every iteration give good results. This reduces μ from 1000 to ~ 0.01 in 40 iterations, which is approximately the number of iterations necessary for convergence. Once μ is fixed we determine the value of τ according to Table 1. However, to ensure stability of the Picard iteration (34) we must require an upper bound of τ_{max} . In our numerical tests we have used $\tau_{max} = 0.5$. The number of Picard iterations in our algorithm are set to 1.

There is also a need for an initial approximation of the constants $\mathbf{c}$. When the image only has 2 phases the initial values for $\mathbf{c}$ is not important, the algorithms converge to the same solution even if the $\mathbf{c}$ values are far from the true ones. For more than 2 phases we need a good approximation for the initial $\mathbf{c}$ values, and this is achieved using a simple isodata approach [4]. During the iterations we can update the $\mathbf{c}$ values using (24). However, this should not be done too early in the process. This is because (24) will give a poor estimate of the $\mathbf{c}$ values unless the level set function ϕ is close to piecewise constant.

This gives us the following algorithm:

Algorithm 3 (*MBO-Newton method*)

- Find initial $\mathbf{c}^0$ values and set $\tau = 0.5$, $\mu = 1000$.
for $n = 1 : n_0$
 - Solve

$$\frac{\phi_i^{n+\frac{1}{2}} - \phi^n}{\tau d} = \beta D_i \cdot \left(\frac{D_i \phi_i^{n+\frac{1}{2}}}{|\nabla \phi_i^n|} \right) - \frac{1}{d} (u(\phi_i^n, \mathbf{c}^n) - u_0) \frac{\partial u}{\partial \phi}(\phi_i^n, \mathbf{c}^n). \quad (42)$$

– Set

$$\phi^{n+\frac{1}{2}} = \frac{1}{d} \sum_{i=1}^d \phi_i^{n+\frac{1}{2}}. \quad (43)$$

– Solve ϕ^{n+1} from

$$\phi^{n+1} - \phi^{n+1/2} = -\frac{\tau}{\mu} W'(\phi^{n+1}). \quad (44)$$

– update $\mathbf{c}^n$ according to (24)

– Update μ and τ through $\mu = 0.75 \cdot \mu$ and $\tau = \min(0.5, \mu\sigma_0)$.

end

Above σ_0 is the upper bound for τ/μ as given in Table 1 and the iteration number n_0 is chosen to be 40 for all the tests we have done. The cost for solving (44) is very cheap. It normally only takes three or four Newton iterations of (40) to get a rather accurate solution for it.

In the rest of this section we will try to explain how the algorithm works. That is, we will try to illustrate how (42) and (43) smooth the level set function ϕ under the restriction that u must be close to u_0 . Whereas, (44) evolves ϕ towards integer values.

In order to show how (42) and (43) are evolving the level set function we will run the algorithm without (44). We also keep $\mathbf{c}$ fixed.

To make the example and visualization as simple as possible we have chosen to use a 1-dimensional signal. Thus instead of a complete image we look at a row of an image. As the initial signal u_0 and the initial level set ϕ^0 we take the noisy step function shown in Figure 1a). Clearly we can see that this function contains 4 levels or phases. Thus we want the level set function to converge towards a function which is close to u_0 , but smoother due to the use of the total variation term. In Figure 1b) we have shown the ϕ function after convergence. From the figure we clearly see that ϕ has converged into a smoothed function containing 4 different levels. It is important to notice that these levels are not 1,2,3 and 4 since we have removed (44).

The parameter β controls the regularization. Thus if we choose β too big we will get a ϕ function that is too smooth, see Figure 2a). On the other hand if we choose β too small we will get a ϕ function that is too noisy, see Figure 2b).

In order to show how (44) is evolving the level set function towards integer values we will run the algorithm without (42) and (43). We have chosen a linear function as shown in Figure 3a) as the initial level set function ϕ^0 . Since we want (44) to force the function values towards the nearest integer we want ϕ to converge into a staircase function, and as Figure 3b) shows, this is exactly what happens.

We now run the entire algorithm, i.e. we combine (42), (43) and (44). We want the level set function ϕ to converge towards $\phi = i$ in Ω_i , $i = 1, 2, \dots, N$, and as Figure 4 shows, this is exactly what happens.

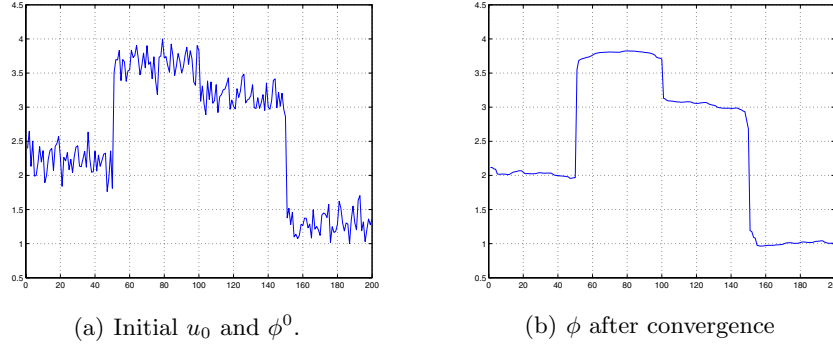


Fig. 1. (42) and (43) evolves the level set ϕ towards a smoothed function containing 4 different levels. β is here set to 0.2.

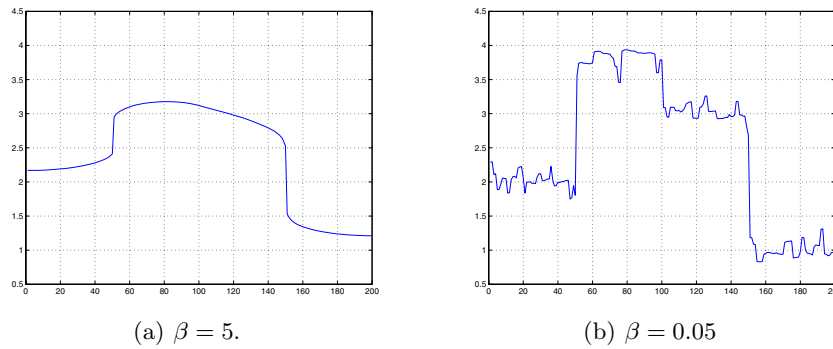


Fig. 2. If we increase β too much we will get a ϕ function which is too smooth. On the other hand if we reduce β too much we will get a ϕ function which is too noisy.

Remark 2. If we take $u = \phi, u_0 = \phi_0$ and only iterate between (42) and (43), then this gives a fast implementation for the ROF total variation denoising algorithms of [15].

Remark 3. Compared with the algorithm of [19], we have replaced the MBO-projection [12] by the solving of (44). The MBO-project is enforcing the constrain so “brutally” that the final results depends on the time step size used for (42)-(43). When it is replaced by (44), the cost for the computation is not increased much and the constraint is also enforced properly by reducing the penalization parameter and the time step according to Table 1.

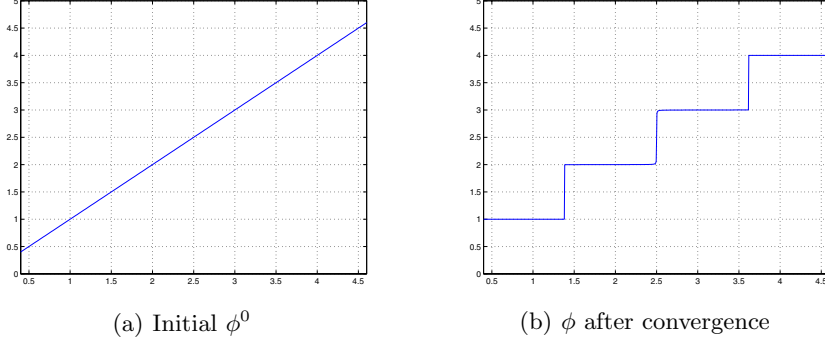


Fig. 3. (44) forces the function values towards the integers and ϕ to converge to a stair function.

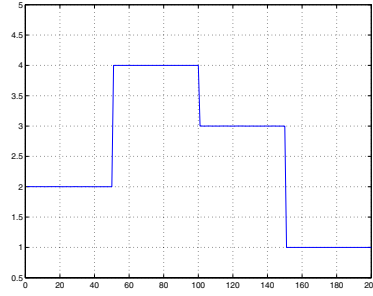


Fig. 4. When the entire algorithm is applied, the level set function ϕ converge towards $\phi = i$ in Ω_i , $i = 1, 2, \dots, N$.

6 Numerical Experiments

In this section we validate our algorithm with numerical experiments for real applications. We consider only two-dimensional cases and restrict ourself to gray-scale images, but the schemes can handle any dimension and can be extended to vector-valued images as well. Synthesized images, natural images and an MR image are evaluated.

The algorithm is as described in section 5 and the advantage of this algorithm is that the only parameter which has to be chosen is the regularizer β . This means that for all images presented in this section we have set the initial $\tau = 0.5$ and the reduction factor $\mu = 0.75$. It might be possible that other images require different values for τ and μ , but we have not experienced this.

All implementations are done in Matlab, and as the initial ϕ function we use the input image u_0 , scaled between one and the number of phases:

$$\phi_0(x) = 1 + \frac{u_0(x) - \min u_0}{\max u_0 - \min u_0}(N - 1), \quad (45)$$

where N is the number of phases. All tests are run on on a 2.8GHz Pentium 4 processor.

In the first example we illustrate a 2-phase segmentation on a real car plate. The size of the image is 370×465 pixels, and the CPU time of the segmentation is 26 s. To challenge the segmentation we add Gaussian distributed noise to the real image and use the polluted image in Figure 5a) as the input data.

To demonstrate the effect of the regularization parameter β we show a number of segmentations with different β values. We see that for $\beta = 0.5$ we have a very good segmentation of the noisy car plate. For smaller β values we are not able to remove the noise, and for larger β values we regularize too much and remove details from the image.

In the next example we show a segmentation of the noisy star image in Figure 6a). The size of the image is 92×98 pixels, and the CPU time for the segmentation is 2.1 s. The star image consists of four different phases, and as Figure 6c) shows the algorithm separates these phases very well. We have also shown the initial and final ϕ function. As Figure 6b) shows, the initial ϕ function is the image scaled between one and four, i.e. the number of phases. After the algorithm has converged the ϕ function only contains four levels, see Figure 6d).

In our next example segmentation of a MR image is demonstrated. The MR image in Figure 7 is available at <http://www.bic.mni.mcgill.ca/brainweb/>. The size of the image is 296×400 pixels and the CPU time for the segmentation is 35 s. These realistic MRI data are used by the neuro imaging community to evaluate the performance of various image analysis methods in a setting where the truth is known. For the image used in this test the noise level is 7% and the non-uniformity intensity level of the RF-puls is 20%.

There are three tissue classes that should be identified; phase 1: cerebrospinal fluid, phase 2: gray matter, phase 3: white matter, and in Figure 8 we have compared the results from our algorithm with the exact phases. We have not depicted the background phase here. We see that we have lost some details, due to the presence of noise.

In Figure 9 we show the results from a 4-phase segmentation of a noisy synthetic image containing 3 objects. The size of the image is 100×100 pixels, and the CPU time for the segmentation is 2.5 s. This is the same image as Chan and Vese used to examine their multiphase algorithm [3, 22]. We see that the algorithm captures the circle and the curved object perfectly, however there is a problem with the triangle. We have a misclassification of the boundary of the triangle. This is probably due to the fact that we regularize directly on the ϕ function in (12). The jump on the boundary of the triangle is twice of the jump on the boundary of the other two objects, see Figure 9 d), and

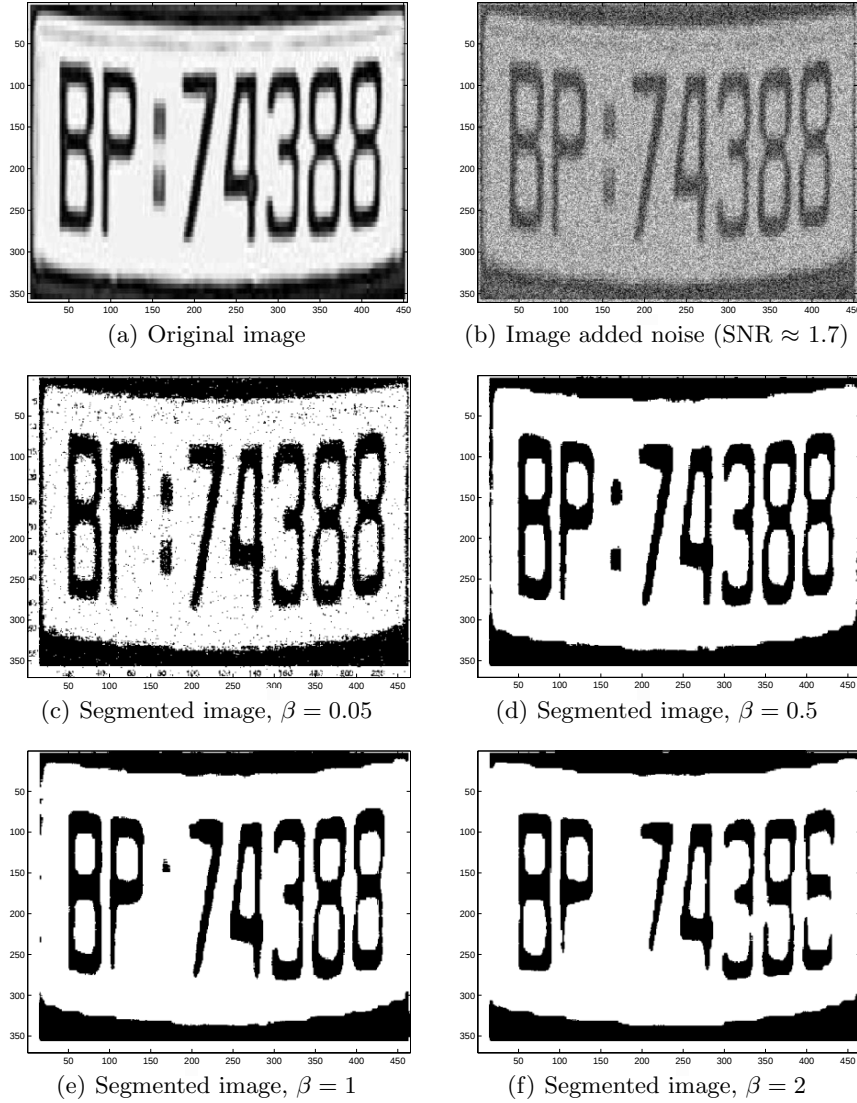


Fig. 5. The regularization parameter β controls the length of the boundary. For $\beta = 0.5$ we have a good segmentation of the car plate. For smaller β we are not able to remove the noise, and for larger β we regularize too much and remove details from the image.

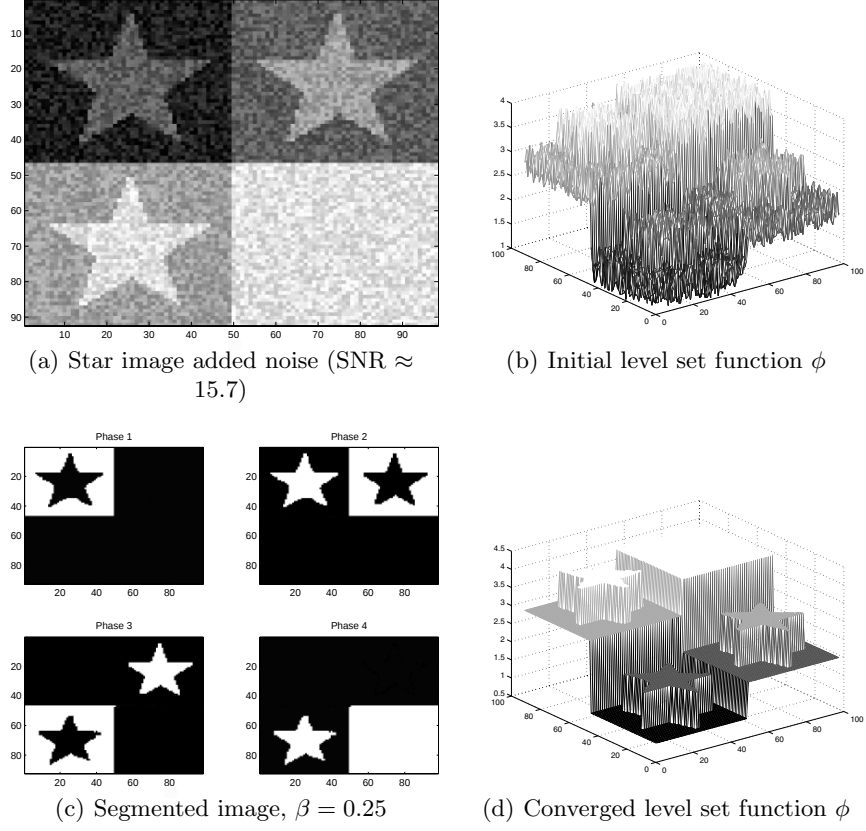


Fig. 6. 4 phase segmentation of a noisy star image.

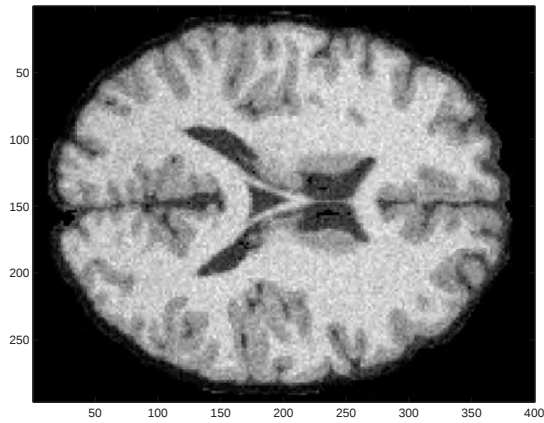


Fig. 7. MRI image with a change in the intensity values going from left to right caused by the non-uniform RF-pulse.

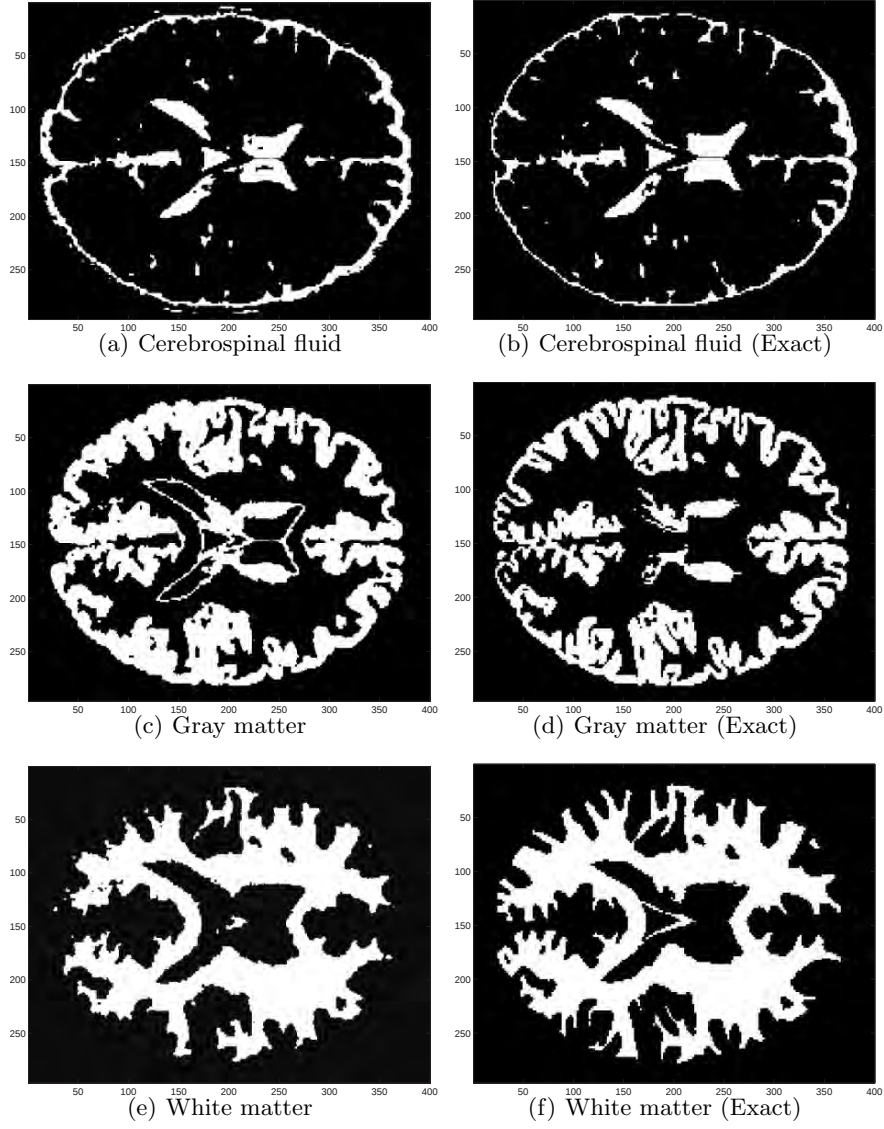


Fig. 8. Comparison of the exact segmentation of a MRI phantom and the results using our algorithm ($\beta = 0.14$).

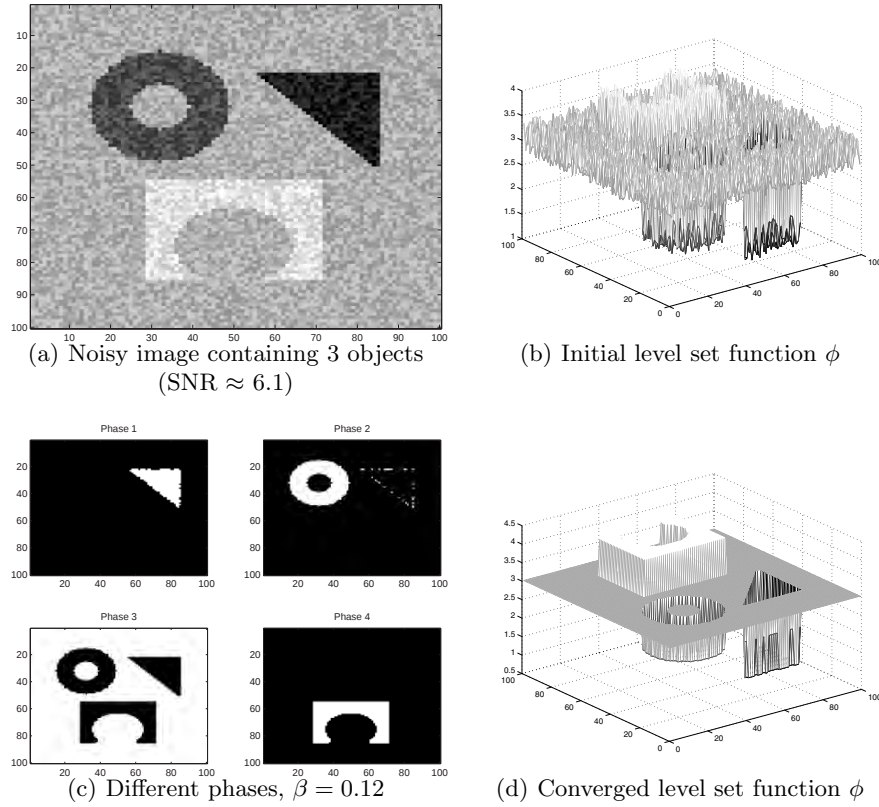


Fig. 9. 4-phase segmentation of a noisy synthetic image containing 3 objects.

the regularization probably “punishes” this jump too hard. Thus in future work we might have to consider to regularize directly on the characteristic functions, in the same manner as in [8].

In our final example we present a two phase segmentation of a real picture of a plane, Figure 10a). The size of the image is 176×101 pixels, and the CPU time for the segmentation is 2.6 s. As before we have added noise to challenge the segmentation, see Figure 10b). We show a number of segmentations with different β values. We see that for $\beta = 0.15$ we have a very good segmentation of the plane. For smaller β values the edges are too noisy, and for larger β values we regularize too much and remove details from the image.

7 Conclusion

Due to the special structure of the PCLSM, we propose a special method to deal with the constraint. In order to have this method to work, we need

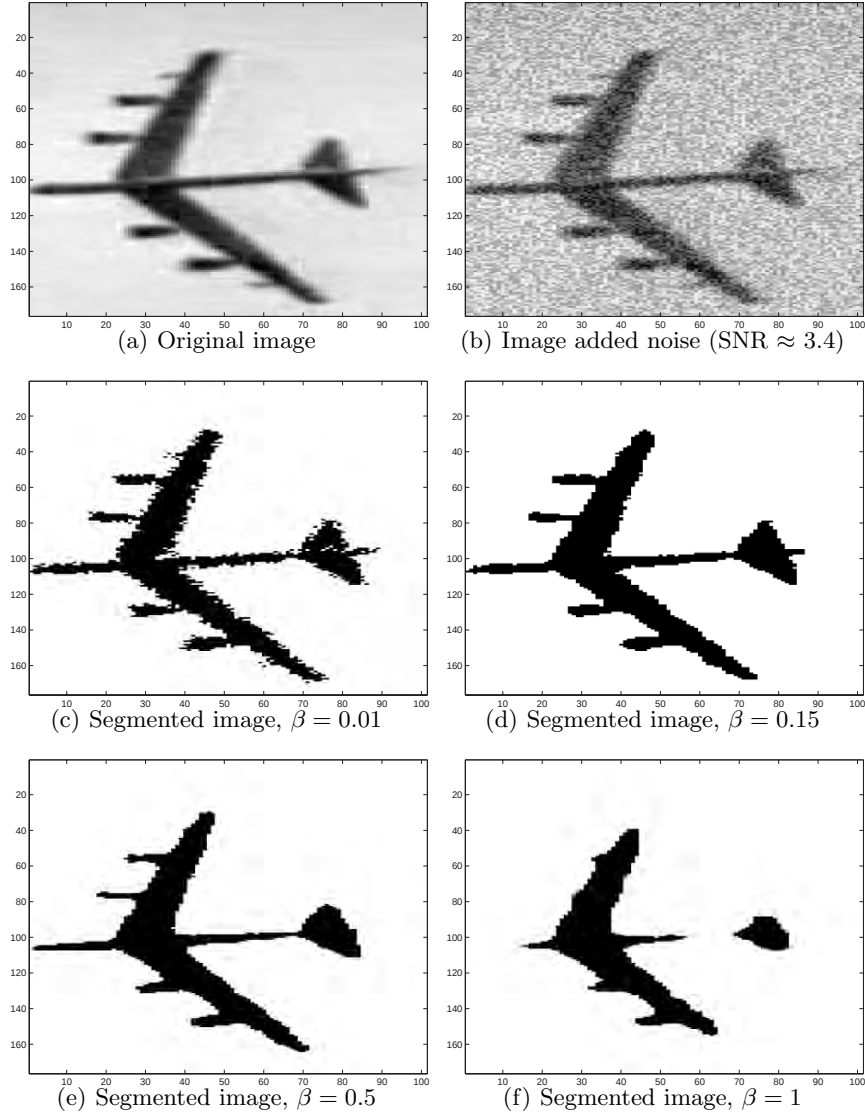


Fig. 10. Different segmentations of a noisy plane image. For $\beta = 0.5$ we have a good segmentation of the plane. For smaller β values the edges are too noisy, and for larger β values we regularize too much and remove details from the image.

to choose the time step τ and the penalization parameter μ to satisfy some inequality. By doing this, we have a very cost efficient way to enforce the constraint. Application to image segmentation is tested in this work. The convergence is fast. Compared with the other fast methods of [19, 20], we do not need good initial values and the algorithm is nearly parameter “free”. It is easy to find values for τ and μ that are good for most of the tested images.

Applications of this idea for PCLSM to inverse problems and multiphase motion problems have also been tested in [21, 7]. Those results show the applicability of the PCLSM for a class of shape identification problems.

References

1. D. Barash. Nonlinear diffusion filtering on extended neighborhood. *Appl. Numer. Math.*, 52(1):1–11, 2005.
2. D. Barash, T. Schlick, M. Israeli, and R. Kimmel. Multiplicative operator splittings in nonlinear diffusion: from spatial splitting to multiple timesteps. *J. Math. Imaging Vision*, 19(1):33–48, 2003.
3. T. F. Chan and L. A. Vese. Image segmentation using level sets and the piecewise constant mumford-shah model. Technical report, CAM Report 00-14, UCLA, Math. Depart., April 2000. revised December 2000.
4. V. Dias and F. R. Threshing using the ISODATA clustering algorithm. *IEEE Trans. Systems Man Cybernet.*, 10(11):771–774, 1980.
5. G. Chung and L. A. Vese. Energy minimization based segmentation and denoising using a multilayer level set approach. In *Energy minimization methods in computer vision and pattern recognition, proceedings lecture notes in computer science, vol. 3757*, pages 439–455, Springer-Verlag, Berlin, 2005.
6. F. Gibou and R. Fedkiw. A fast hybrid k-means level set algorithm for segmentation. Technical report, Stanford Technical Report, 2002.
7. H. Li and X.-C. Tai. Piecewise constant level set methods (PCLSM) for interface problems. Technical report, UCLA, Applied Mathematics, 2006.
8. J. Lie, M. Lysaker, and X.-C. Tai. A variant of the levelset method and applications to image segmentation. *UCLA CAM03-50*, 2003 (to appear in Math. Comp.).
9. A. Litman. Reconstruction by level sets of n -ary scattering obstacles. *Inverse Problems*, 21:131–152, 2005.
10. T. Lu, P. Neittaanmaki, and X.-C. Tai. A parallel splitting up method and its application to Navier-Stoke equations. *Applied Mathematics Letters*, 4:25–29, 1991.
11. T. Lu, P. Neittaanmaki, and X.-C. Tai. A parallel splitting up method for partial differential equations and its application to Navier-Stokes equations. *RAIRO Math. Model. and Numer. Anal.*, 26:673–708, 1992.
12. B. Merriman, J. K. Bence, and S. J. Osher. Motion of multiple functions: a level set approach. *J. Comput. Phys.*, 112(2):334–363, 1994.
13. D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math*, 42:577–685, 1989.
14. S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

15. L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
16. C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia. A level set model for image classification. *IJCV*, 40(3):187–197, 2000.
17. J. Shen. Gamma-convergence approximation to piecewise constant Mumford-Shah segmentation. Tech. Rep. CAM05-16, UCLA Dep. Math, 2005.
18. B. Song and T.F. Chan. Fast algorithm for level set segmentation. *UCLA CAM report 02-68*, 2002.
19. X.-C. Tai, O. Christiansen, P. Lin, and I. Skjælaaen. Fast implementation of piecewise constant level set methods. *International Journal of Computer Vision*, to appear.
20. X.-C. Tai and C. Yao. Fast piecewise constant level set methods (PCLSM) with Newton updating. *UCLA CAM 05-52*, 2005.
21. X.-C. Tai and H. Li. Piecewise constant level set methods (PCLSM) for elliptic inverse problems. Cam-report-05-59, UCLA, Applied Mathematics, 2005.
22. L. A. Vese and T. F. Chan. A new multiphase level set framework for image segmentation via the Mumford and Shah model. *International Journal of Computer Vision*, 50:271–293, 2002.
23. J. Weickert, B.H. Romeny, and M.A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE TRans. Image Process.*, 7:398–409, 1998.
24. J. Weickert and G. Kühne. Fast methods for implicit active contour models. In *Geometric level set methods in imaging, vision, and graphics*, pages 43–57. Springer, New York, 2003.
25. H.-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127(1):179–195, 1996.

The Multigrid Image Transform

Paul M. de Zeeuw

CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands. E-mail:
`Paul.de.Zeeuw@cwi.nl`

Summary. A second order partial differential operator is applied to an image function. To this end we consider both the Laplacian and a more general elliptic operator. By using a multigrid operator known from the so-called approximation property, we derive a multiresolution decomposition of the image without blurring of edges at coarser levels. We investigate both a linear and a nonlinear variant and compare to some established methods.

Key words: Elliptic multigrid image transform, gradient pyramids, Laplace equation, Laplacian pyramids, Laplacian multigrid image transform, lifting scheme, multigrid methods, multiresolution, steerable pyramids, wavelets.

1 Introduction

In a more or less parallel development the idea of multiresolution has become an important instrument both in the field of signal processing and in the field of numerical methods for the solution of partial differential equations (PDEs). With respect to the latter we allude to the *multigrid* type of method which solves discretized elliptic, parabolic and hyperbolic PDEs as well as integral equations by accelerating a basic iterative solution process through adequate coarse grid corrections [5, 14]. A historical overview of the development including a list of pioneering papers is given by Wesseling [26].

Terzopoulos [23] was the first to apply multigrid for image analysis. More recently, the use of multigrid for image processing purposes has been proposed by Acton [1], Kimmel et al. [16], Shapira [20], Ke Chen et al. [9], Bruhn et al. [6] and others. However, its use is restricted to the efficient solution of partial differential equations (typically diffusion and Euler-Lagrange equations) which could also be achieved by other means.

De Zeeuw (this author) started to use multigrid operators as an intrinsic and indissoluble part of the so-called *multigrid image transform* [13]. In this

scheme, first a second order partial differential operator is applied to an image function followed by a pyramidal decomposition using typical multigrid operators. The case of isotropic homogeneous diffusion (Poisson) provides an example that leads to a linear multiresolution scheme. It can be applied successfully with respect to image fusion [13].

In the present paper we consider a general elliptic operator but we focus on the isotropic inhomogeneous diffusion operator, with coefficients in the fashion of Perona and Malik [18, 19]. It leads to a nonlinear multiresolution scheme. A future application of the new scheme might be in image fusion using a nonlinear multiresolution decomposition implying a multisource segmentation.

The paper is organized as follows. After a recapitulation on multigrid in Section 2 we discuss the multigrid image transform in Section 3. In particular we consider one that is associated with the Laplacian (leading to a linear multiresolution scheme) and one that is associated with a more general elliptic partial differential operator (leading to a nonlinear multiresolution scheme). We show results of the transforms in Section 4 and compare to other multiresolution schemes amongst which a nonlinear one by Heijmans and Goutsias [15]. We end up with concluding remarks.

2 Recapitulation on Multigrid

A prohibitive problem with the solution of large (non)linear systems of equations is that the number of arithmetic operations involved is more than linearly proportional to the number of unknowns. For example, the complexity of the direct solution of large sparse linear systems is still quadratic even when exploiting the structured sparsity. Also the fill-in demands more than proportional storage. Such systems arise after the discretization of PDEs on a spatial grid. For special PDEs, e.g. Poisson problems, considerable efficiency can yet be achieved, for an overview see e.g. Botta et al. [4].

Multigrid is a numerical class of methods which tackles the complexity problem head-on by representing and solving a problem and its derivations on a sequence of increasingly coarser (finer) grids. Nowadays extensive literature is available on multigrid. We merely point to Brandt [5], Hackbusch [14], Wesseling [26] and (more recent) to Trottenberg et al. [24] and Shapira [20].

Here we recapitulate particular items that we need for the multigrid transform to be discussed from an article by De Zeeuw (this author) on a robust multigrid algorithm for the numerical solution of (scalar) diffusion and convection-diffusion problems [10]. The algorithm has been implemented and exists by the name of MGD9V. Tests demonstrate its (optimal) complexity for a wide range of problems known to be difficult to solve. It employs a set of rectangular and increasingly coarser grids (vertex-centered):

$$\Omega_n \supset \Omega_{n-1} \supset \dots \Omega_k \supset \dots \supset \Omega_0.$$

The grids are described as follows:

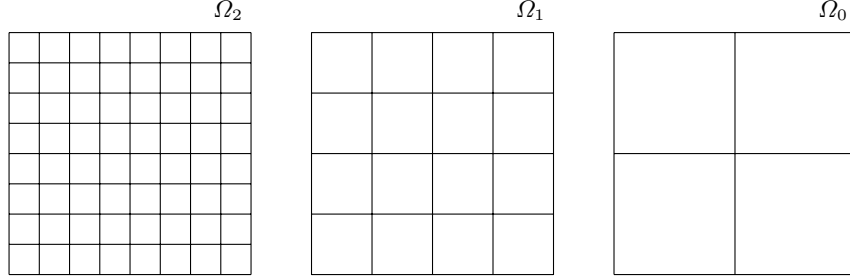


Fig. 1. Example sequence of increasingly coarsened grids used in multigrid (vertex-centered)

$$\Omega_k \equiv \{(x_i, y_i) \mid x_i = o_1 + (i-1)h_k, y_i = o_2 + (j-1)h_k\} \quad (1)$$

where (o_1, o_2) is the origin and $h_{k-1} = 2h_k$. See Figure 1 for an example. $S(\Omega_k)$ denotes the linear space of real-valued functions on Ω_k

$$S(\Omega_k) = \{g_k \mid g_k : \Omega_k \rightarrow \mathbb{R}\},$$

where $g_k \in S(\Omega_k)$ is called a *grid-function*. The algorithm is intended for the solution of linear systems resulting from the 9-point discretization of the following general linear second-order elliptic partial differential equation in two dimensions:

$$Lu \equiv -\nabla \cdot (D(x)\nabla u(x)) + b(x) \cdot \nabla u(x) + c(x)u(x) = f(x) \quad (2)$$

on a bounded domain $\Omega \subset \mathbb{R}^2$ with suitable boundary conditions. $D(x)$ is a positive definite 2×2 matrix function and $c(x) \geq 0$. It is assumed that the discretization of (2) is performed by a finite element or finite volume technique, leading to

$$L_n \bar{u}_n = f_n \quad (3)$$

where

$$L_n : S(\Omega_n) \rightarrow S(\Omega_n) \quad (4)$$

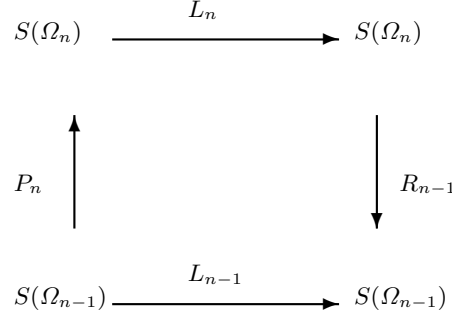
is the discretization of L and $f_n \in S(\Omega_n)$ is the discretization of f . Grid-function $\bar{u}_n$ is the solution that is looked for. The solution algorithm uses *sawtooth multigrid cycles*, that is, a smoother is applied after the coarse grid correction (CGC). Let u_n be an approximation of $\bar{u}_n$. The CGC at level k reads:

$$r_k = f_k - L_k u_k; \quad (5)$$

$$r_{k-1} = R_{k-1} r_k; \quad (6)$$

$$\text{solve (approximately) } L_{k-1} e_{k-1} = r_{k-1}; \quad (7)$$

$$\tilde{u}_k = u_k + P_k e_{k-1}. \quad (8)$$

**Fig. 2.** Diagram of Galerkin approximation

It is immediately followed by:

$$\tilde{u}_k = \text{SMOOTH}(f_k, L_k, \tilde{u}_k). \quad (9)$$

In MGD9V the particular choice for $\text{SMOOTH}()$ is *Incomplete Line LU factorization* (for a description see [11] and the references mentioned there). The grid transfer operators are defined as follows.

$$R_{k-1} : S(\Omega_k) \rightarrow S(\Omega_{k-1}), \quad k = n, \dots, 1 \quad (10)$$

is the restriction operator that transfers the residual from the grid Ω_k onto the coarser grid Ω_{k-1} , and

$$P_k : S(\Omega_{k-1}) \rightarrow S(\Omega_k), \quad k = 1, \dots, n \quad (11)$$

is the prolongation operator that interpolates and transfers a correction for the solution from the coarser towards the finer grid. The operator L_{k-1} is defined by the sequence of operations

$$L_{k-1} \equiv R_{k-1} L_k P_k, \quad k = n, \dots, 1 \quad (12)$$

known as the *Galerkin coarse grid approximation*. One cycle of sawtooth multigrid is defined by application of (5)–(9) for $k = n$. A recursion enters at stage (7). The system of equations at this stage is approximated by applying again the above cycle, but now at level $k - 1$. (At level 0 mere smoothing is performed).

The diagram of Figure 2 illustrates the coherence of the afore mentioned operators. We choose the restriction to be the transpose of the prolongation

$$R_{k-1} = P_k^T, \quad k = n, \dots, 1. \quad (13)$$

Hence, once P_k has been chosen, R_{k-1} and L_{k-1} follow automatically. One actually computes the coarse grid matrix of L_{k-1} . Note that by (13) the

possible (anti)symmetry of L_k is maintained on the coarser grid. Further, it has been proved [10] that when L_k is a conservative discretization of L and P_k interpolates a constant function exactly, then the Galerkin approximation L_{k-1} is conservative as well. In the case of e.g. the Poisson equation and discretization by bilinear finite elements, bilinear interpolation is the natural choice for P_k . This case is discussed in Section 3.2. In the case of discontinuous diffusion coefficients a far more sophisticated choice is required [10]. This case is discussed in Section 3.3.

Adiabatic Boundary Conditions

At the boundaries of Ω one often assumes vanishing Neumann boundary conditions. At Ω_n we discretize them in a conservative fashion, e.g. by using bilinear finite elements. The following statements can all be derived from [10]. The boundary conditions inherited by L_k , $0 \leq k < n$, remain vanishing Neumann ones. All L_k , $0 \leq k \leq n$ have a singular matrix and therefore the L_k^{-1} do not exist. However, systems of type $L_k u_k = g_k$ can still be solved, provided that g_k is in the range of L_k . A sufficient and necessary condition for the latter is proved to be that the sum of elements of g_k vanishes. The said discretization warrants this condition for $k = n$. Further, it is proved that $R_{k-1}g_k$ inherits the condition. It follows that the multigrid algorithm in [10] is able to solve the described systems iteratively, even though the matrix L_n is singular. The solution u_k is unique up to a constant (grid-function).

3 The Multigrid Image Transform

3.1 Introduction

So far, we have recapitulated how a multigrid method solves large linear systems of equations arising from discretized PDEs in a very efficient manner based on a recursive procedure. However, the current section is not about multigrid solution methods, but about image *transforms* involving multigrid operators. The exploits of Section 2 provide some necessary tools for the transforms to be discussed. Another tool that we need is the multigrid approximation operator

$$E_k : S(\Omega_k) \rightarrow S(\Omega_k), \quad k = 1, \dots, n \quad (14)$$

which is defined as:

$$E_k \equiv L_k^{-1} - P_k L_{k-1}^{-1} R_{k-1}, \quad k = 1, \dots, n. \quad (15)$$

It is associated with the so-called *approximation property*. Under a certain regularity of the boundary value problem (2), a discretization (3) by (bilinear) finite elements, and P_k is bilinear interpolation, it can be shown that (see Hackbusch [14, §6.3]):

$$\|E_k\|_2 \leq Ch_k^2 \quad (16)$$

where h_k is the mesh-size of Ω_k and $\|\cdot\|_2$ is the Euclidean norm on $S(\Omega_k)$. This operator plays an important role in convergence proofs in multigrid theory. In [13] it has been proposed to let E_k serve a practical purpose as well. There it is introduced as a high-pass filter in a multiresolution scheme: the *multigrid image transform*[13]. The transform reads as follows. Let u_n be an image, defined as a grid-function on $S(\Omega_n)$. Then compute grid-function $f_n = L_n u_n$, for the definition of L_n see (2) and (3). Note that this is contrary to finding a solution u_n for given f_n , which was the problem stated in Section 2. An important example for L_n is the discretized Laplacian operator, this is discussed in Section 3.2. Let

$$f_k \equiv R_k f_{k+1}, \quad k = n-1, \dots, 0 \quad (17)$$

then we define the *multigrid image transform* or *multigrid image decomposition* as follows

$$\begin{cases} a_0 &= L_0^{-1} f_0, \\ d_k &= E_k f_k, \quad k = 1, \dots, n. \end{cases} \quad (18)$$

The a_k are called *approximations* and the d_k are called *details*. The reconstruction counterpart reads:

$$a_k = P_k a_{k-1} + d_k, \quad k = 1, \dots, n. \quad (19)$$

Regarding (3), (10)–(12), (15), (17)–(19) it follows that

$$L_k a_k = f_k, \quad k = 0, \dots, n.$$

which implies that the reconstruction (19) with respect to the decomposition (18) is a perfect one. The proof can be found in a previous paper [13].

As with other multiresolution methods, manipulations of the detail coefficients d_k may allow for a better tackling of image processing problems.

Adiabatic Boundary Conditions Revisited

Under these boundary conditions E_k is meaningful, even though it is not defined in the strict sense. It can be proved that if g_k is in the range of L_k then $R_{k-1}g_k$ is in the range of L_{k-1} and therefore $E_k g_k$ can still be applied. Again, the result is unique up to a constant (grid-function).

3.2 The Laplacian Multigrid Image Transform

Laplacian

Firstly, we consider the case of both isotropic and homogeneous diffusion which boils down to the use of the Laplacian operator $-\Delta$. Let L_n be the discretization on the grid Ω_n (uniform and rectangular). If discretized by

means of bilinear finite elements (or volumes) it gives rise to the following 3×3 stencil (or mask) for meshsize 1:

$$L_n \sim \frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{bmatrix}. \quad (20)$$

Bilinear Prolongation

Under the assumption of (13), the prolongation must satisfy an accuracy condition, in order to obtain mesh-size independent rate of multigrid convergence. Such an accuracy condition is increasingly stringent for higher orders of the PDE, for more details see [5, 14, 26]. Here, bilinear interpolation satisfies the accuracy condition for the second order PDE. This interpolation amounts to taking an equal average of solution-values at neighbouring coarse-grid points, see Figure 3 for an illustration. At the grid-points of the fine grid that coincide

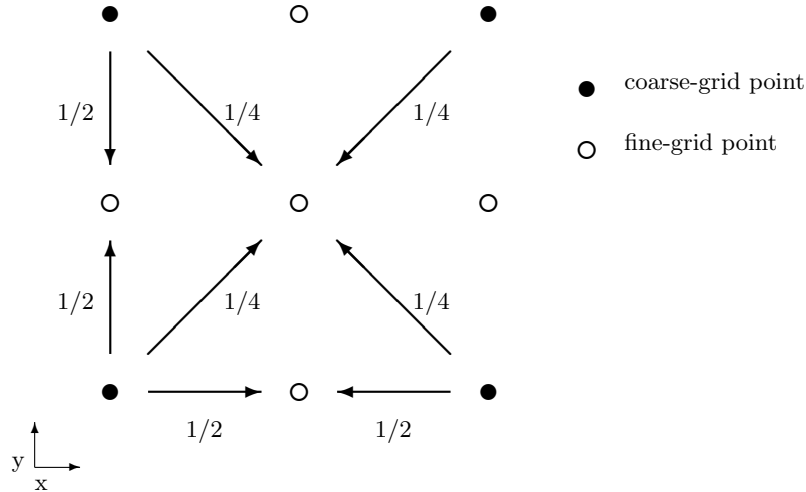


Fig. 3. Bilinear prolongation.

with the coarse grid we take identical values. The bilinear prolongation can also be denoted by the stencil

$$P_k \sim \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}. \quad (21)$$

This stencil shows the non-zero values of the fine-grid function generated by the prolongation of a coarse-grid function which equals 1 at one point and 0 elsewhere. Because of (13), the same stencil also represents the chosen restriction operator.

Ease of Implementation

With the prolongation and restriction thus chosen the Laplacian stencil (20) is invariant on the coarser grids. That is, all L_k produced by (12) turn out to be represented by the same stencil on the subsequently coarser grids $S(\Omega_k)$, $0 \leq k < n$. We assume adiabatic boundary conditions which are also retained. The proof can be derived from [10].

Through this foreknowledge the multigrid method can be simplified greatly with respect to its implementation. It is not necessary to perform (12) explicitly as we already know the outcome both in the interior and at the boundaries. Another simplification lies in the choice of the basic iterative method (also known as smoother or relaxation method). With the above Laplacian stencil one can resort to simple and vectorizable smoothers like e.g. damped Jacobi. Moreover, the method becomes economical with computer memory as storage of matrices and their decompositions is not required.

3.3 The Elliptic Multigrid Image Transform*Matrix-dependent Prolongations and Restrictions*

We recall the elliptic operator (2) defined in Section 2. We add that the positive definite tensor D is allowed to be discontinuous across an interface Γ in the interior of Ω . Obviously, definitions of coefficients in the fashion of Perona and Malik allow for this to happen. Let L_n be the discretization on Ω_n (uniform and rectangular grid) by means of bilinear finite elements (or volumes). When D is strongly discontinuous, multigrid with bilinear prolongation becomes excruciatingly slow: the number of iterative cycles necessary to obtain a fixed reduction of r_n becomes prohibitively large. The explanation is as follows. Let $n = n(x)$ be the normal at Γ . Then, as has been argued by Alcouffe et al. [2], continuity of $n \cdot (D\nabla u)$ instead of continuity of ∇u should be the underlying assumption for interpolation. This leads to jump conditions that need to be satisfied across interfaces. Only in the (special) case that the diffusion coefficient D is continuous, it follows that ∇u is continuous as well and the use of bilinear interpolation is justified. For an illustrative one-dimension example on interface problems see Hackbusch [14, §10.3.1]. The right assumption that $n \cdot (D\nabla u)$ is continuous leads to the remedy of operator-dependent prolongations (and restrictions). Figure 4 provides an *in situ* illustration of a biased prolongation, satisfying a jump condition for the case that the diffusion coefficient is negligible in the shaded region. One notes the obvious differences with Figure 3.

In [10] a matrix-dependent prolongation operator has been proposed, able to handle both the case of (dominant) advection and interface problems at the same time. Here we give a brief outline of the operator. At each level k the (black box) multigrid algorithm derives the necessary information on the operator coefficients from the matrix L_k (this explains the adjective “matrix-dependent”). The grid Ω_k is split into four disjoint sub-grids as follows:

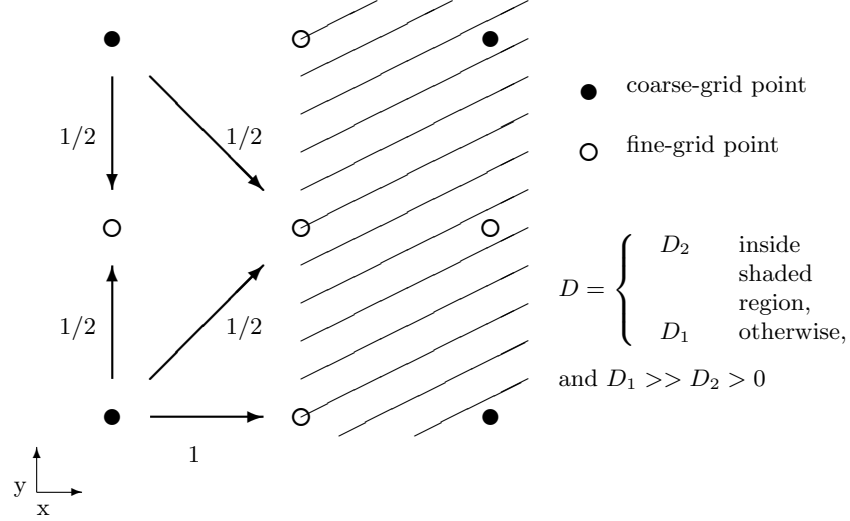


Fig. 4. Example of biased prolongation.

$$\begin{aligned}
 \Omega_{k,(0,0)} &\equiv \Omega_{k-1}, \\
 \Omega_{k,(1,0)} &\equiv \{(x + h_k, y) \in \Omega_k \mid (x, y) \in \Omega_{k-1}\}, \\
 \Omega_{k,(0,1)} &\equiv \{(x, y + h_k) \in \Omega_k \mid (x, y) \in \Omega_{k-1}\}, \\
 \Omega_{k,(1,1)} &\equiv \{(x + h_k, y + h_k) \in \Omega_k \mid (x, y) \in \Omega_{k-1}\},
 \end{aligned}$$

where h_k is the mesh-size of grid Ω_k . We proceed as follows.

1. At the fine-grid points in $\Omega_{k,(0,0)}$, we simply adopt the values on Ω_{k-1} .
2. Let $\xi \in \Omega_{k,(1,0)}$ be a point where we have to interpolate a coarse grid correction. It is by definition located on a horizontal grid-line between two neighbouring points at Ω_{k-1} . Locally, we decompose the matrix L_k in its symmetric and antisymmetric part. The symmetric part is presumed to correspond with diffusion and the zeroth order term, the antisymmetric part with convection. We reconstruct the various operator coefficients at ξ and apply essentially one-dimensional interpolation. The interpolation coefficients are stored.
3. Let $\xi \in \Omega_{k,(0,1)}$ be a point where we have to interpolate a coarse grid correction. We interpolate as above, but now on a vertical grid-line of Ω_{k-1} .
4. At the fine-grid points in $\Omega_{k,(1,1)}$, we solve the homogeneous equation (with respect to L_k) to obtain the correction.
5. Now that P_k has been defined (and therefore R_{k-1} as well) we compute L_{k-1} according to (12) at the next coarser grid and we repeat the whole process above for level $k - 1$ ($k > 0$).

Definition

Summarizing, the *elliptic multigrid image transform* is defined by (17)–(18), through the elliptic operator L and its discretization L_n (see (2) and (3)), through the matrix-dependent P_k and (12)–(13). The Laplacian multigrid image transform of Section 3.2 is a particular example of this transform.

Implementation

The implementation of the actual computation of L_{k-1} according to (12) with the above matrix-dependent P_k is far from trivial. The implementation of a highly robust smoother like e.g. incomplete line LU factorization is also not a trivial matter, but it is what the multigrid method wants due to the discontinuous diffusion coefficients. For these reasons, the general elliptic multigrid image transform is more intricate than the Laplacian one. Nevertheless, the necessary work is of low and linear complexity. (The stencils L_k do not grow on the coarser grids but remain 3×3 just like L_n .)

4 Comparative Results

Perona and Malik Type Diffusivity

For experiments with the elliptic multigrid transforms we limit ourselves to the case of no convection and no zeroth order term. With respect to the diffusivity we consider diffusion which is again isotropic but inhomogeneous. It boils down to the use of the operator $-\nabla \cdot (D \nabla u)$ where D is scalar-valued, not a tensor (several possibilities exist for D as tensor as pointed out by Weickert [25]). Perona and Malik [18, 19] have reasoned that intra-region smoothing should occur preferentially over inter-region smoothing. The diffusion is chosen locally as a function of the magnitude of the gradient of the image function

$$D(x) = g(|\nabla u(x)|^2). \quad (22)$$

With respect to the function g we opt here for the following:

$$g(s) = \frac{1}{\sqrt{1+s}} \quad (23)$$

see Aubert et al. [3, §3.3.1] for a full motivation. In the context of the Perona-Malik model this gives better smoothing in the tangential direction than in the normal direction.

Discretized, this diffusivity expresses the coupling that exists between points in the image. By means of (12) this coupling is transferred to coarser grids. The matrix-dependent grid transfer operators secure that weak (strong) couplings remain weak (strong). Therefore, as with time integration, the diffusivity helps to preserve edges (but now on coarsened grids).

Experiments

We apply both the Laplacian and the elliptic multigrid transform with the above diffusion operator, both with adiabatic boundary conditions, to the grayscale image at the top of Figure 5. We compare with the results of well-known linear multiresolution schemes as wavelets [17] (see Figure 5) and Laplacian pyramids [7], gradient pyramids [8] and steerable pyramids [21] (see Figure 6).

Further, in Figure 7, we compare with the results of what we refer to as the “maxmin-lifting scheme”. This scheme is a nonlinear version of the lifting scheme [22] involving quincunx grids. It is defined by intertwined use of the nonlinear max- and min-lifting schemes by Heijmans and Goutsias [15]. The max-lifting scheme has the property that it preserves local maxima over several scales. The min-lifting scheme has a similar property with respect to local minima. An implementation of the maxmin-lifting scheme can be found through [12]. Clearly, Figure 7 depicts the least blurring of edges on subsequently coarsened grids.

Efficiency

Table 1 shows CPU times consumed on a 2.16 GHz processor by a few selected multiresolution schemes (decomposition plus reconstruction) on grids with different dimensions. The costs of the schemes appear to be within the same

Table 1. CPU seconds consumed by multiresolution schemes

Grid	Daubechies 4	maxmin-lifting	elliptic MG
256×256	0.43	0.45	0.30
512×512	0.74	0.94	0.79
1024×1024	2.40	3.82	3.08

range. Moreover, the measurements accord with the claimed computational complexities.

5 Concluding Remarks

New multiresolution schemes have been investigated, based on an image transform by a discretized elliptic partial differential operator and use of a multigrid operator, leading to pyramidal representations. Depending on the differential operator, the scheme is linear or nonlinear. The linear scheme (Laplacian multigrid image transform) is easy to implement, rapidly converging and economical with storage. An example of the nonlinear scheme (elliptic multigrid



Fig. 5. Top: original image. Middle and bottom row show approximations on subsequently coarsened grids (from left to right). Middle row: Haar wavelet decomposition. Bottom row: wavelet decomposition by Daubechies 4.



Fig. 6. Approximations on subsequently coarsened grids (from left to right). Top row: Laplacian pyramid. Middle row: gradient pyramid. Bottom row: steerable pyramid (6 bands).

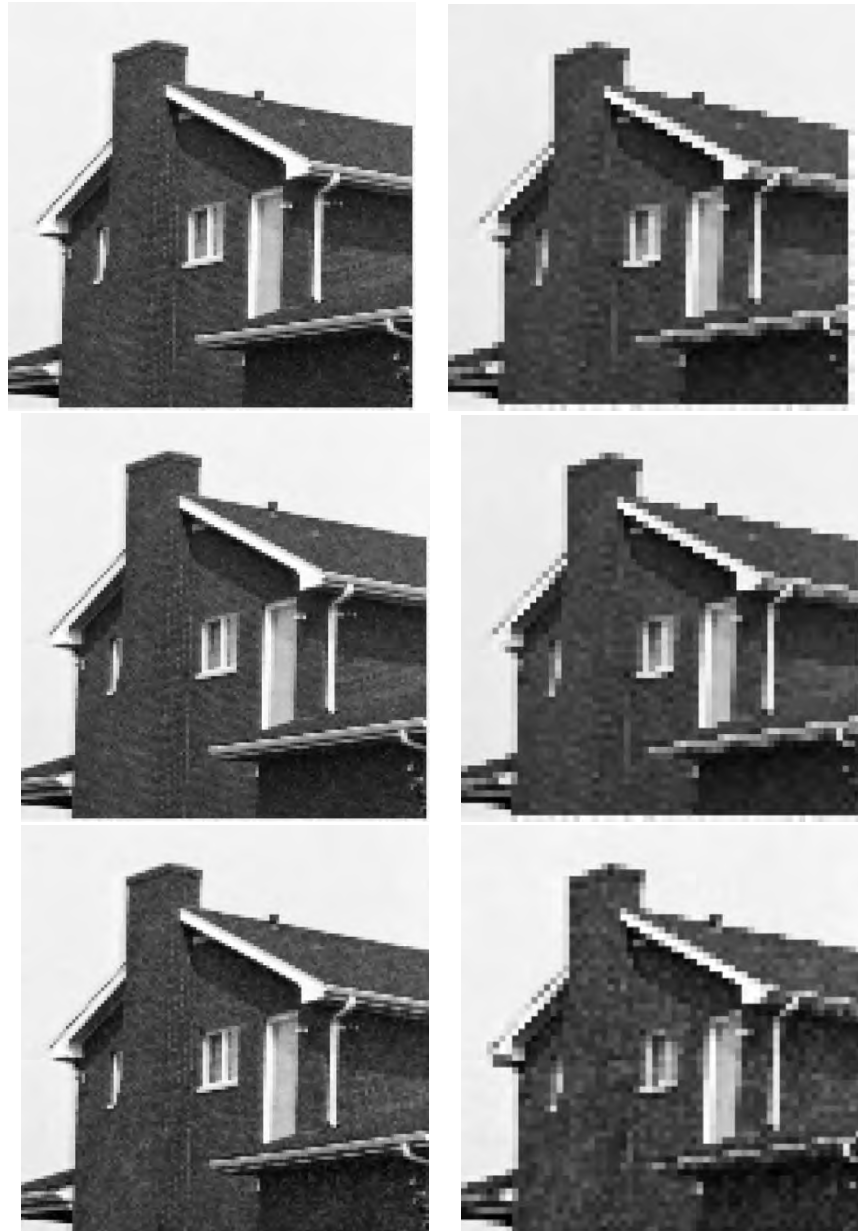


Fig. 7. Approximations on subsequently coarsened grids (from left to right). Top row: Laplacian multigrid image transform. Middle row: elliptic multigrid image transform. Bottom row: maxmin-lifting scheme.

image transform) based on Perona and Malik type diffusivity has been developed. Though more intricate than the linear scheme, the complexity remains low and linear. A comparison with several well-known and established linear multiresolution schemes has been made, but also with a nonlinear lifting scheme. The latter scheme and both multigrid image transforms appear to be in the same league with respect to preservation of edges at coarser grids. The elliptic multigrid image transform appears to have a slight edge over the nonlinear lifting scheme.

So far, we have considered mere scalar diffusion. A diffusion tensor leading to anisotropic (tensor) diffusion filters [25] with special spatial regularization properties could be a topic for future research. Another future topic could be image fusion, as the elliptic multigrid image transform appears to relate to segmentation.

References

1. S. T. Acton. Multigrid anisotropic diffusion. *IEEE Trans. Image Process.*, 7(3):280–291, 1998.
2. R. E. Alcouffe, A. Brandt, J. E. Dendy, and J. W. Painter. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. *SIAM J. Sci. Statist. Comput.*, 2:430–454, 1981.
3. G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing, Partial Differential Equations and the Calculus of Variations*, volume 147 of *Applied Mathematical Sciences*. Springer Verlag, New York, 2002.
4. E. F. F. Botta, K. Dekker, Y. Notay, A. van der Ploeg, C. Vuik, F. W. Wubs, and P. M. de Zeeuw. How fast the Laplace equation was solved in 1995. *J. Applied Numerical Mathematics*, 24:439–455, 1997.
5. A. Brandt. Multi-level adaptive techniques (mlat) for partial differential equations: ideas and software. In J. R. Rice, editor, *Mathematical Software III*, pages 277–318. Academic Press, New York, 1977.
6. A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. Discontinuity-preserving computation of variational optic flow in real-time. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 279–290. Springer-Verlag, Berlin Heidelberg, 2005.
7. P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.
8. P. J. Burt and R. J. Kolczynski. Enhanced image capture through fusion. In *Proceedings Fourth International Conference on Computer Vision*, pages 173–182, Los Alamitos, California, 1993. IEEE Computer Society Press.
9. K. Chen and X.-C. Tai. A nonlinear multigrid method for curvature equations related to total variation minimization. Report 05-26, UCLA CAM, 2005.
10. P. M. de Zeeuw. Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. *J. Comput. Appl. Math.*, 33:1–27, 1990.
11. P. M. de Zeeuw. Chapter 14: Multigrid and advection. In C. B. Vreugdenhil and B. Koren, editors, *Numerical Methods for Advection-Diffusion Problems*,

- volume 45 of *Notes on Numerical Fluid Mechanics*, pages 335–351. Vieweg, Braunschweig, 1993.
12. P. M. de Zeeuw. A toolbox for the lifting scheme on quincunx grids (lisq). CWI Report PNA-R0224, Centrum voor Wiskunde en Informatica, Amsterdam, 2002. <http://www.cwi.nl/ftp/CWIreports/PNA/PNA-R0224.pdf>.
 13. P. M. de Zeeuw. A multigrid approach to image processing. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 396–407. Springer-Verlag, Berlin Heidelberg, 2005.
 14. W. Hackbusch. *Multi-Grid Methods and Applications*, volume 4 of *Computational Mathematics*. Springer-Verlag, Berlin, 1985.
 15. H. J. A. M. Heijmans and J. Goutsias. Multiresolution signal decomposition schemes. part ii: Morphological wavelets. *IEEE Trans. Image Process.*, 9(11):1897–1913, 2000.
 16. R. Kimmel and I. Yavneh. An algebraic multigrid approach for image analysis. *SIAM J. Sci. Comput.*, 24(4):1218–1231, 2003.
 17. S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Pattern Anal. Mach. Intell.*, 11(7):674–693, 1989.
 18. P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Pattern Anal. Mach. Intell.*, 12(7):629–639, 1990.
 19. P. Perona, T. Shiota, and J. Malik. Anisotropic diffusion. In Bart M. ter Haar Romeny, editor, *Geometry-Driven Diffusion in Computer Vision*, volume 1 of *Computational Imaging and Vision Series*, pages 73–92. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.
 20. Y. Shapira. *Matrix-Based Multigrid: Theory and Applications*. Kluwer Academic Publishers, Boston, 2003.
 21. E. P. Simoncelli and W. T. Freeman. The steerable pyramid: a flexible architecture for multi-scale derivative computation. In *Proceedings of the IEEE International Conference on Image Processing*, pages 444–447. IEEE Signal Processing Society, 1995.
 22. W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.
 23. D. Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8:129–139, 1986.
 24. U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, London, 2001.
 25. J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, Stuttgart, 1998.
 26. P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley & Sons Ltd., Chichester, 1991.

Minimally Stochastic Schemes for Singular Diffusion Equations

Bernhard Burgeth¹, Joachim Weickert¹, and Sibel Tari²

¹ Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science, Bldg. E2 4, Saarland University, 66041 Saarbrücken, Germany. E-mail: {burgeth,weickert}@mia.uni-saarland.de, url: <http://www.mia.uni-saarland.de>

² Department of Computer Engineering, Middle East Technical University, 06531 Ankara, Turkey. E-mail: stari@metu.edu.tr, url: <http://www.ceng.metu.edu.tr>

Summary. Total variation (TV) and balanced forward-backward (BFB) diffusion are prominent examples of singular diffusion equations: Finite extinction time, the experimentally observed tendency to create piecewise constant regions, and being free of parameters makes them very interesting. However, their appropriate numerical treatment is still a challenge. In this paper a minimally stochastic approach to these singular equations is presented. It is based on analytical solutions of two-pixel signals and stochastic rounding. This introduces regularisation via integer arithmetic and does not require any limits on the diffusivity. Experiments demonstrate the favourable performance of the proposed probabilistic method.

Key words: Randomisation, total variation, balanced forward-backward diffusion, singular diffusivity

1 Introduction

1.1 The Setting

Initiated with the work of Perona and Malik [11] nonlinear diffusion filters have become an important tool for image processing. The basic setting of diffusion filtering is as follows. An initial image $f : \Omega \rightarrow \mathbb{R}$ given on an two-dimensional domain $\Omega \subset \mathbb{R}^2$ is subjected to an evolutionary process governed by the following partial differential equation (PDE) with Neumann boundary conditions:

$$\begin{aligned} \partial_t u &= \operatorname{div} (g(|\nabla u|) \cdot \nabla u) && \text{on } \Omega \times (0, \infty) \\ u(x, 0) &= f(x) && \text{for all } x \in \Omega \\ \partial_n u(x, t) &= 0 && \text{for all } x \in \partial\Omega \times (0, \infty) \end{aligned} \tag{1}$$

with outward normal derivative ∂_n on the image domain boundary $\partial\Omega$. This evolution process creates more simplified versions $u(\cdot, t)$ of f the larger the time parameter t is. One can steer this process to achieve edge preservation and intraregional smoothing by specifying the diffusivity g as a nonnegative and decreasing function of $|\nabla u|$.

Many nonlinear diffusion filters rely on bounded diffusivities [6, 11]. However, in recent years unbounded diffusivities that became singular at zero have received special attention [8, 2, 10, 7]. In numerical experiments these filters create cartoon-like, piecewise constant images. In this paper we will focus on two choices for the diffusivity g , both rendering the corresponding PDE singular: The specification

$$g(|\nabla u|) = \frac{1}{|\nabla u|} \quad (2)$$

gives rise to the total variation (TV) diffusion [2, 9]. The TV diffusion filter is associated with TV regularisation if a penaliser $\Psi(|\nabla u|^2) = 2|\nabla u|$ is used [14]. Among the most interesting properties of TV diffusion are finite extinction time [3], certain shape-preserving qualities [4], and equivalence results to TV regularisation for one-dimensional signals [5, 12].

The specification

$$g(|\nabla u|) = \frac{1}{|\nabla u|^2} \quad (3)$$

generates the so-called balanced forward-backward diffusion (BFB), [10]. For this type of diffusion actual edge enhancement occurs. Note that neither TV nor BFB diffusion require any filter parameter tuning. Generalisations of this diffusion filters replacing the square by a positive exponent p also have been considered in [1, 16].

Numerical difficulties are the price to be paid for the appealing properties of TV or BFB diffusion: In order to apply classical finite difference schemes, one needs bounded diffusivities. This is achieved by replacing $|\nabla u|$ by $\sqrt{|\nabla u|^2 + \epsilon^2}$ in the denominators of (2) and (3). However, the time step size in explicit finite difference schemes is reciprocal to bounds on the diffusivity function to ensure stability, and the condition numbers of system matrices emerging from absolutely stable semi-implicit or implicit schemes are increasing functions of such bounds. This entails high computational complexity and/or potential amplification of numerical errors. Moreover the bounded diffusivity introduces the unpleasant side effect that blurring artefacts occur and theoretical considerations for singular diffusion filters are no longer applicable.

An alternative that does not require a regularised diffusivity is described in [15]: In a two-pixel setting analytic solutions of systems of ordinary differential equations associated with a spatial discretisation of the singular PDE are employed for numerical evaluation. In [18] the same idea of utilising analytical solutions of ODE-systems has been put to work successfully in the more complicated framework of four pixels. Both approaches lead to absolutely stable

explicit scheme at the expense of having conditional consistency only: When the product of the time step size and the diffusivity becomes large, a linear diffusion process is approximated. This means that for increasing time step sizes, more and more blurring artifacts arise.

The goal of the present paper is to address this problem. By introducing an approximation that allows only integers as grey values, we bound the gradient away from zero: The employed one-sided discretisation $|\nabla u|_{i,j}$ of $|\nabla u|$ in (12) entails that either $|\nabla u|_{i,j} = 0$, which can be treated separately, or $|\nabla u|_{i,j} \geq \frac{1}{\sqrt{2}h}$ with grid size h . This implies that the discrete approximations $g_{i,j}$ for the diffusivity are bounded by $\sqrt{2}h$ in the case of TV-diffusion and by $2h^2$ for BFB-diffusion. Hence we are allowed to use larger time step sizes without visual deterioration than in the conventional 2- or 4-pixel schemes.

Since diffusion is an inherently continuous process that should also be allowed to proceed in infinitesimally small steps, it is not possible to design a satisfying diffusion scheme that uses integer arithmetic in a deterministic framework by conventional rounding. As a remedy, we introduce a minimal amount of randomisation in the spirit of [13]. It is realised by a stochastic rounding procedure which introduces fluctuations that are small enough to be invisible, but large enough to have a beneficial regularising effect.

The paper is structured as follows: The two-pixel scheme based on an analytic solution of a system of ODEs is introduced in the first part of the next section. In its second part the analytic two-pixel scheme is randomised by stochastic rounding leading to the proposed minimally stochastic method. Numerical experiments in section 3 show the favourable performance of the minimally stochastic approach when compared to the purely deterministic method. Section 4 with a short summary and remarks about ongoing work completes the paper.

2 Schemes Based on Two Pixel Interaction

2.1 Deterministic Approach

We will start our investigation with the simplest possible case. We are considering a one-dimensional version of (1) discretised by two pixels with homogeneous Neumann boundary conditions:

$$f = (f_1, f_2), \quad \text{resp.,} \quad u = (u_1, u_2)$$

A space-discrete, but time-continuous scheme for (1) is then given by

$$\begin{aligned} \dot{u}_1 &= \frac{g_{1+\frac{1}{2}}}{h^2} (u_2 - u_1) \\ \dot{u}_2 &= \frac{-g_{1+\frac{1}{2}}}{h^2} (u_2 - u_1) \end{aligned}$$

with initial conditions $u_i(0) = f_i$, $i = 1, 2$. Here the discrete approximants g_1 and g_2 of the diffusivity g at pixel 1 and 2 are calculated using dummy pixels $u_0 := u_1$ and $u_3 := u_2$ yielding $g_{1+\frac{1}{2}}$ by

$$g_{1+\frac{1}{2}} := \frac{g_1 + g_2}{2}$$

In general, first order differences are approximated by standard central differences $\frac{1}{2h}(u_{i-1} - u_{i+1})$ with grid size h .

We assume that $g_{1+\frac{1}{2}}$ is independent of time, that is, constant with respect to t in this coupled system of ordinary differential equations. In order to decouple this system of ODEs we introduce $w_1(t) = u_2(t) - u_1(t)$ and $v_1(t) = u_2(t) + u_1(t)$, in fact

$$\begin{pmatrix} w_1(t) \\ v_1(t) \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} \quad (4)$$

Then the function w_1 satisfies the linear first order ODE

$$\dot{w}_1 = \frac{2}{h^2} g_{1+\frac{1}{2}} w_1$$

which is readily solved to give

$$w_1(t) = \exp\left(-\frac{2}{h^2} g_{1+\frac{1}{2}} \cdot t\right) w_1(0)$$

For the sum $v_1(t)$ we obtain the ODE

$$\dot{v}_1(t) = 0$$

yielding $v_1(t) = v_1(0) = u_2(0) + u_1(0)$ for all $t \geq 0$. With this at our disposal solving the equation system (4) gives

$$\begin{aligned} \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} &= \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} w_1(t) \\ v_1(t) \end{pmatrix} \\ &= \begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} + \frac{1}{2} \left(1 - \exp\left(-\frac{2t}{h^2} g_{1+\frac{1}{2}}\right)\right) (u_2(0) - u_1(0)) \begin{pmatrix} 1 \\ -1 \end{pmatrix} \end{aligned}$$

Considering now n -pixel signals we may apply this reasoning to any pair of pixels u_i and u_{i+1} . Thus we obtain

$$\begin{aligned} u_i(t) &= u_i(0) + \frac{1}{2} \left(1 - \exp\left(-\frac{2t}{h^2} g_{i+\frac{1}{2}}\right)\right) (u_{i+1}(0) - u_i(0)) \\ u_{i+1}(t) &= u_{i+1}(0) - \frac{1}{2} \left(1 - \exp\left(-\frac{2t}{h^2} g_{i+\frac{1}{2}}\right)\right) (u_{i+1}(0) - u_i(0)) \end{aligned}$$

or in its time discrete variant after k iterations with time step size τ

$$\begin{aligned} u_i^{k+1} &= u_i^k + \frac{1}{2} \left(1 - \exp \left(-\frac{2\tau}{h^2} g_{i+\frac{1}{2}}^k \right) \right) (u_{i+1}^k - u_i^k) \\ u_{i+1}^{k+1} &= u_{i+1}^k - \frac{1}{2} \left(1 - \exp \left(-\frac{2\tau}{h^2} g_{i+\frac{1}{2}}^k \right) \right) (u_{i+1}^k - u_i^k) \end{aligned} \quad (5)$$

However, this ensures interaction between the two neighbouring pixels u_i^k and u_{i+1}^k only, pixel u_{i-1}^k , say, is not involved. In order to overcome this problem we consider also a shifted version of the signal, follow the procedure indicated above and average the two signal versions in an additive operator splitting (AOS) approach [17]: We allow for diffusion between u_i^k and u_{i+1}^k with time step size 2τ yielding

$$\tilde{u}_i^{k+1} = u_i^k + \frac{1}{2} \left(1 - \exp \left(-\frac{4\tau}{h^2} g_{i+\frac{1}{2}}^k \right) \right) (u_{i+1}^k - u_i^k) \quad (6)$$

and we enable diffusion between u_i^k and u_{i-1}^k with time step size 2τ by setting

$$\tilde{u}_i^{k+1} = u_i^k - \frac{1}{2} \left(1 - \exp \left(-\frac{4\tau}{h^2} g_{i-\frac{1}{2}}^k \right) \right) (u_i^k - u_{i-1}^k) \quad (7)$$

Then averaging $u_i^{k+1} = \frac{1}{2}(\tilde{u}_i^{k+1} + \tilde{u}_i^{k+1})$ results in

$$\begin{aligned} u_i^{k+1} &= u_i^k + \frac{1}{4} \left(1 - \exp \left(-\frac{4\tau}{h^2} g_{i+\frac{1}{2}}^k \right) \right) (u_{i+1}^k - u_i^k) \\ &\quad - \frac{1}{4} \left(1 - \exp \left(-\frac{4\tau}{h^2} g_{i-\frac{1}{2}}^k \right) \right) (u_i^k - u_{i-1}^k) \end{aligned} \quad (8)$$

The combination of these two steps according to the AOS-framework permits the transport of information throughout the image domain, since it provides a coupling between all pixels. Only this ensures the usefulness of the two-pixel module described in (5), res., in (6) and (7).

Note that a formal first order Taylor expansion w.r.t. τ of the exponential expressions yields the explicit scheme

$$\begin{aligned} u_i^{k+1} &= u_i^k + \frac{\tau}{h^2} g_{i+\frac{1}{2}}^k (u_{i+1}^k - u_i^k) \\ &\quad - \frac{\tau}{h^2} g_{i-\frac{1}{2}}^k (u_i^k - u_{i-1}^k) \end{aligned} \quad (9)$$

The stability of scheme (9) will be destroyed by large diffusivity values. In contrast to that the exponential scheme (8) remains stable, However, as all unconditionally stable explicit schemes, it is only conditionally consistent: If the product of the time step size and the diffusivity becomes large the algorithm turns into simple averaging, and therefore approximates linear diffusion.

In the two-dimensional case of images an analog derivation leads to the scheme

$$\begin{aligned}
u_{i,j}^{k+1} = & u_{i,j}^k + \frac{1}{8} \left(1 - \exp \left(-\frac{8\tau}{h^2} g_{i+\frac{1}{2},j}^k \right) \right) (u_{i+1,j}^k - u_{i,j}^k) \\
& + \frac{1}{8} \left(1 - \exp \left(-\frac{8\tau}{h^2} g_{i-\frac{1}{2},j}^k \right) \right) (u_{i-1,j}^k - u_{i,j}^k) \\
& + \frac{1}{8} \left(1 - \exp \left(-\frac{8\tau}{h^2} g_{i,j+\frac{1}{2}}^k \right) \right) (u_{i,j+1}^k - u_{i,j}^k) \\
& + \frac{1}{8} \left(1 - \exp \left(-\frac{8\tau}{h^2} g_{i,j-\frac{1}{2}}^k \right) \right) (u_{i,j-1}^k - u_{i,j}^k)
\end{aligned}$$

Since we are averaging over twice as many neighbours as in the 1-D case, the weight 4 had been replaced by 8. This scheme is also well-suited for singular diffusivities, it is unconditionally stable and conditionally consistent.

2.2 Minimally Stochastic Approach

We want to construct an integer-valued analog to the process (5), that is, a system

$$\begin{aligned}
u_m^{k+1} &= u_m^k + \omega \\
u_n^{k+1} &= u_n^k - \omega
\end{aligned} \tag{10}$$

where ω can only assume integer values. This warrants that the integer grey values of the initial image remain integer valued during the whole evolution process. As already mentioned conventional rounding is not an feasible option, hence we introduce a form of randomised rounding. This amounts to the design of a randomising module that requires the data of only two pixels as input. Instead of rounding by $[x] = \text{integer part of } x$, this module utilizes a stochastic rounding function $SR : \mathbb{R} \rightarrow \mathbb{Z}$ defined by

$$SR(x) := \begin{cases} [x] & \text{with probability } 1 - |x - [x]| \\ [x] + 1 & \text{with probability } |x - [x]| \end{cases}$$

One finds, for example,

$$SR(2.7) = \begin{cases} 2 & \text{with probability } 0.3 \\ 3 & \text{with probability } 0.7 \end{cases}$$

We employ this random variable to turn (10) with

$$\omega := SR \left[\frac{1}{2} \left(1 - \exp \left(-\frac{2\tau}{h^2} \frac{g_n^k + g_m^k}{2} \right) \right) (u_n^k - u_m^k) \right] \tag{11}$$

into a randomised and integer-valued variant of a 2-pixel scheme.

The regularising effect of the proposed stochastic rounding allows for larger time steps. A standard deterministic rounding would not be appropriate if

the image is piecewise almost flat. In this case deterministic rounding would not permit to diffuse small quantities which would entail unphysical results. Instead we allow for fluctuations of one grey level in magnitude and hereby exploit the insensitivity of the visual system to small changes in greyvalues.

So far the exchange of information between two pixels is ensured. Now the task that remains is to transport the information to other pixels. The idea close at hand would be to use an additive operator splitting like in the deterministic case. However, this would come down to averaging four integer solutions in each pixel, such that there is no guarantee that the result is an integer number again. This is the reason why we use a multiplicative operator splitting for our randomised approach. Since it leads to a sequential application of the randomised two-pixel interactions, integer results are ensured. In the 2-D setting there are 8 different ways of passing through all pixels in a regular order, as is indicated in Fig. 1. Selecting one of these cases, however,

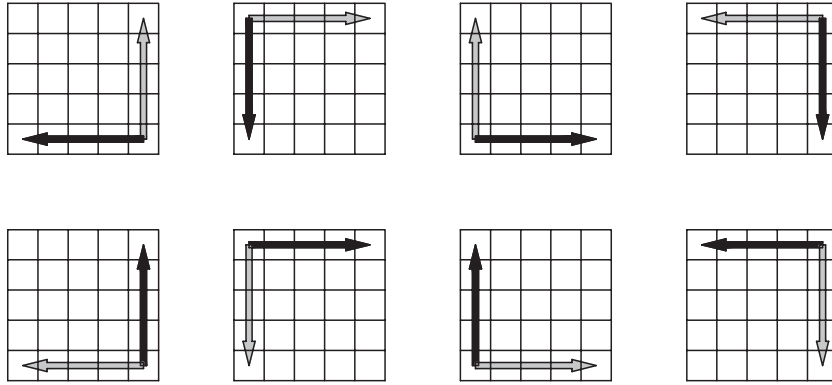


Fig. 1. Extension of the two-pixel-scheme to a 2D-image by applying it to overlapping pairs of pixels. Selection of the starting point and marching directions indicated by black and grey arrows.

would introduce a directional bias for a nonlinear PDE such as TV flow. In order to avoid this problem, we introduce a second randomisation in our algorithm: We randomly choose one of the eight cases which are considered to be equally likely, namely of having probability $\frac{1}{8}$ each. From the numerical point of view the following issues have turned out to be beneficial:

- If the initial data (f_i) are integer valued the scheme in (10) together with 11 produces integer values only.
- Since the diffusivities considered are unbounded the case that $g_m^k = \infty$ or $g_n^k = \infty$ must be accounted for by setting

$$\omega := SR\left(\frac{u_n^k - u_m^k}{2}\right)$$

- From the numerical point of view it is advantageous to compute reciprocal diffusivities $\frac{1}{g_m^k}$ and use the harmonic mean for averaging:

$$\omega = \begin{cases} SR \left\{ \left(1 - \exp \left(-\frac{4\tau}{h^2} \left(\frac{1}{g_n^k} + \frac{1}{g_m^k} \right)^{-1} \right) \right) \frac{u_m^k - u_n^k}{2} \right\}, & \text{for } \frac{1}{g_m^k}, \frac{1}{g_n^k} > 0, \\ SR \left(\frac{u_m^k - u_n^k}{2} \right), & \text{for } \frac{1}{g_m^k} \text{ or } \frac{1}{g_n^k} = 0. \end{cases}$$

It is important to remark that the proposed minimally stochastic scheme produces filtered data consisting of integer values as soon as the initial data are integer valued making it suitable for simple hardware. The scheme also obeys a minimum-maximum-principle since the two-pixel process does. This is an important stability issue.

3 Numerical Experiments

In this section we display some results of numerical experiments to visualise the properties of the deterministic exponential and the minimally stochastic approach. We consider a 256×256 greyvalue medical image and a 128×128 image where in 70% of its pixels the grey value is replaced by an value randomly chosen according to a uniform distribution on $\{0, 1, \dots, 255\}$. For the discretisation of $|\nabla u|$ we used one sided differences:

$$|\nabla u_{i,j}| = \left\{ \frac{1}{2} \left(\left(\frac{u_{i+1,j} - u_{i,j}}{h} \right)^2 + \left(\frac{u_{i,j} - u_{i-1,j}}{h} \right)^2 \right) + \frac{1}{2} \left(\left(\frac{u_{i,j+1} - u_{i,j}}{h} \right)^2 + \left(\frac{u_{i,j} - u_{i,j-1}}{h} \right)^2 \right) \right\}^{\frac{1}{2}} \quad (12)$$

We subject the images to TV-diffusion based on both the deterministic and minimally stochastic two-pixel-scheme. The total diffusion time of 100 is achieved with time step sizes $\tau = 0.01, 0.1, 1$, that is, with 10000, 1000, 100 iterations. The sequence of filtered images indicates clearly the stabilising effect of the randomisation: The minimally stochastic computation allows for about 10 times larger time steps when compared with a deterministic counterpart of the same visual quality. While with a time step size of $\tau = 1$ the deterministic scheme produces an output degraded by fluctuations and blurring effects, the minimally stochastic approach still yields a satisfactory result.

The situation is similar but less pronounced in the case of BFB-diffusion. Here the total diffusion time is 3000 tackled with time step sizes $\tau = 3, 10, 30$ which entails 1000, 300, 100 iterations. Again the regularising effect of the minimally stochastic computation is clearly discernable, however, the gain is now an about three times larger time step in comparison with a qualitatively similar deterministic result.

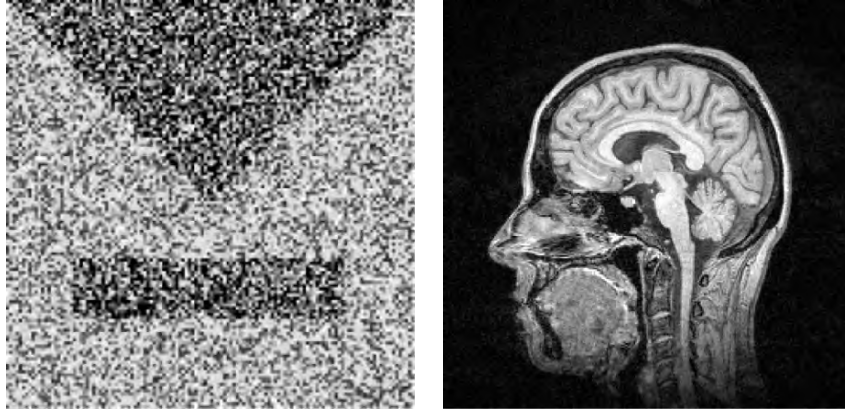


Fig. 2. Test images. *Left:* A 128×128 image polluted with 70% uniform noise. A 256×256 image without additional noise.

The computational gain achieved by the minimally stochastic approach over the deterministic method is documented for both TV- and BFB-diffusion in table 1 10000 iterations each have been performed on a Athlon XP 2.4 Ghz CPU for a grey value image of size 256×256 . One can say that the deterministic and the minimally stochastic scheme are computationally equally costly.

Table 1. CPU time necessary for 10000 iterations performed with the deterministic explicit or minimally stochastic scheme for TV- and BFB-diffusion.

	deterministic	minimally stochastic
TV	7 min 20.6 sec	7 min 28.3 sec
BFB	7 min 25.8 sec	7 min 26.1 sec

4 Conclusion

The usage of singular diffusivities has advantages, like feature preserving qualities and the absence of tuning parameters, for instance. However, numerical intricacies turn the actual calculations into a challenging task. In this paper we introduce a minimally stochastic approach that regularises the singular diffusion filter. It is based on a time-continuous but space-discrete explicit two-pixel-scheme for which an analytical solution can be derived. This two-pixel-scheme receives a random component by employing stochastic rounding. The regularising effect of this randomisation allows for much larger time steps when compared with the deterministic two-pixel-scheme, and for integer valued initial data it can be realised in such a way that only integer arithmetic

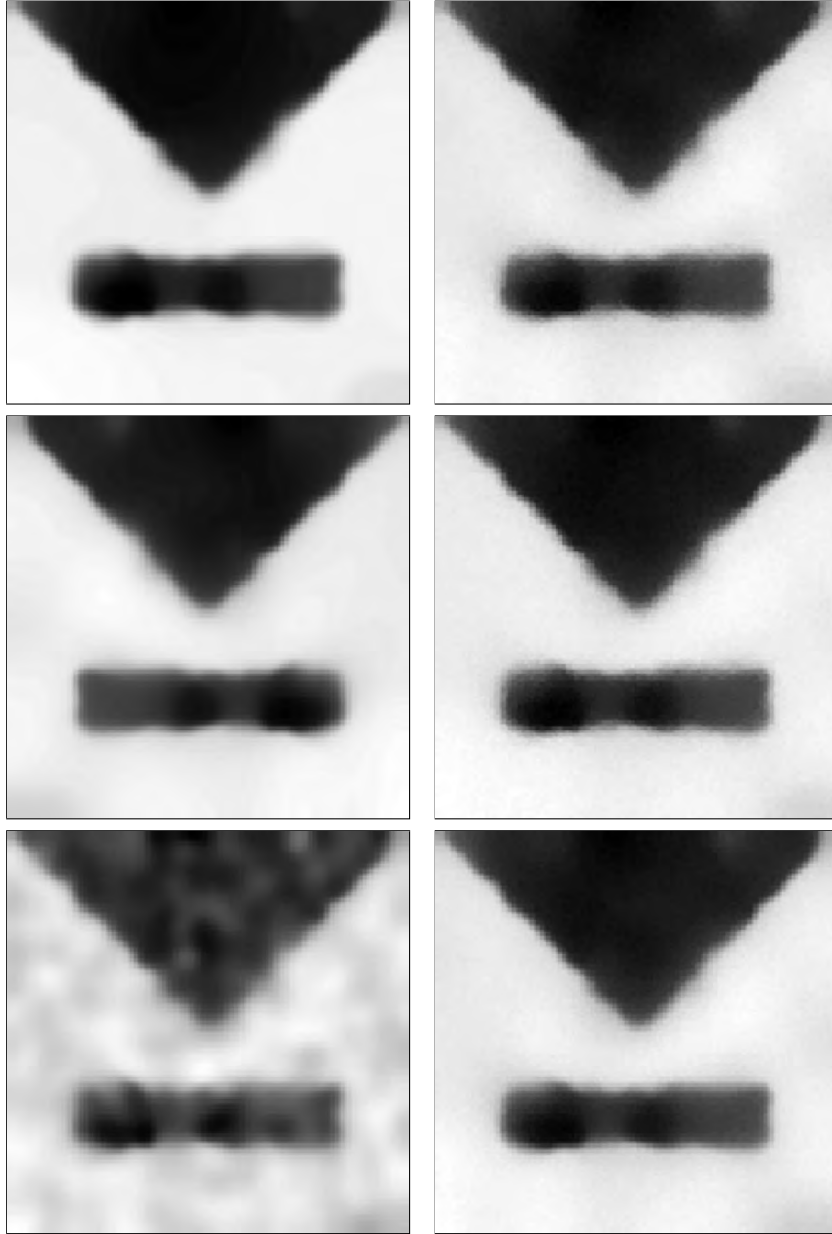


Fig. 3. Comparing deterministic and minimally stochastic computations of TV diffusion filtering with total diffusion time 100.

Left column: Deterministic calculation with explicit scheme.

Right column: Minimally stochastic calculation.

From top to bottom: Time step size $\tau = 0.01, 0.1$, and 1 requiring $10^4, 10^3$, and 10^2 iterations, respectively.

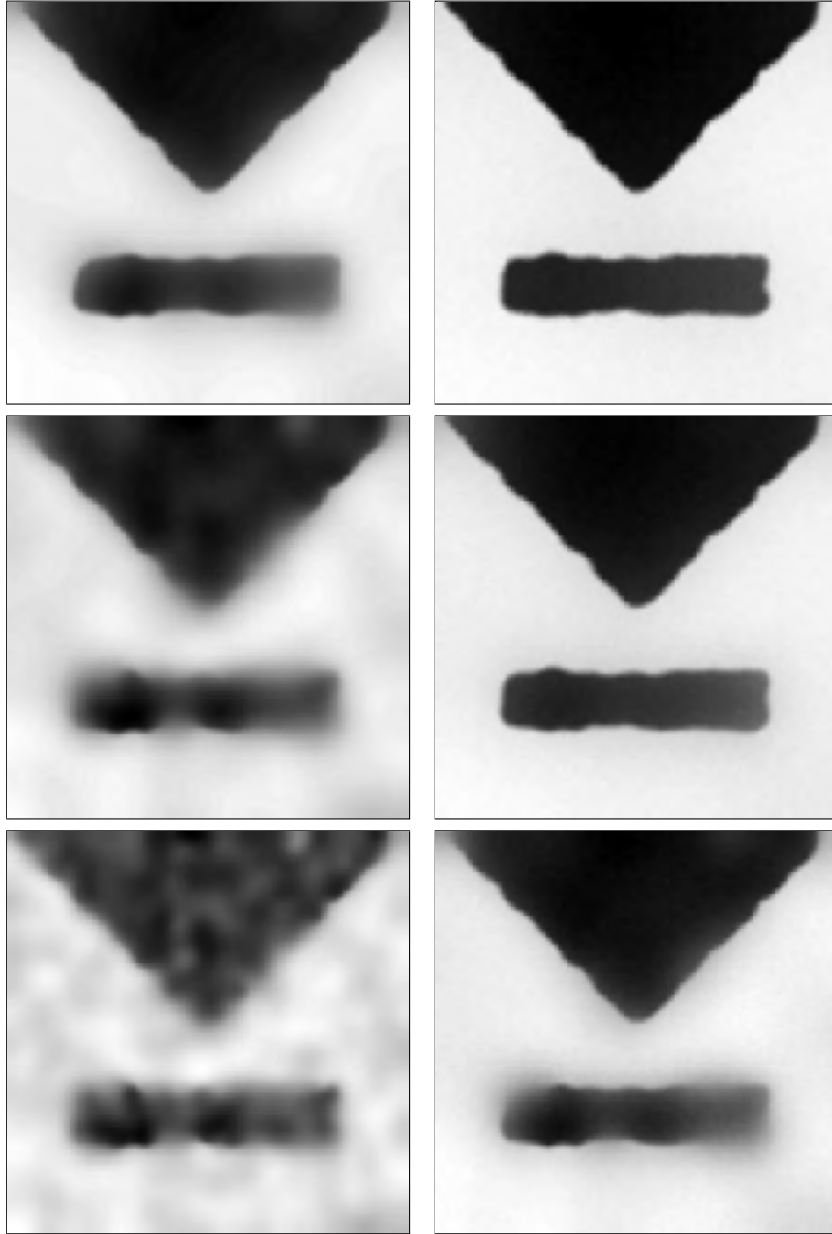


Fig. 4. Comparing deterministic and minimally stochastic computations of BFB diffusion filtering with total diffusion time 3000.

Left column: Deterministic calculation with explicit scheme.

Right column: Minimally stochastic calculation.

From top to bottom: Time step size $\tau = 3, 10$, and 30 requiring 1000, 300, and 100 iterations, respectively.

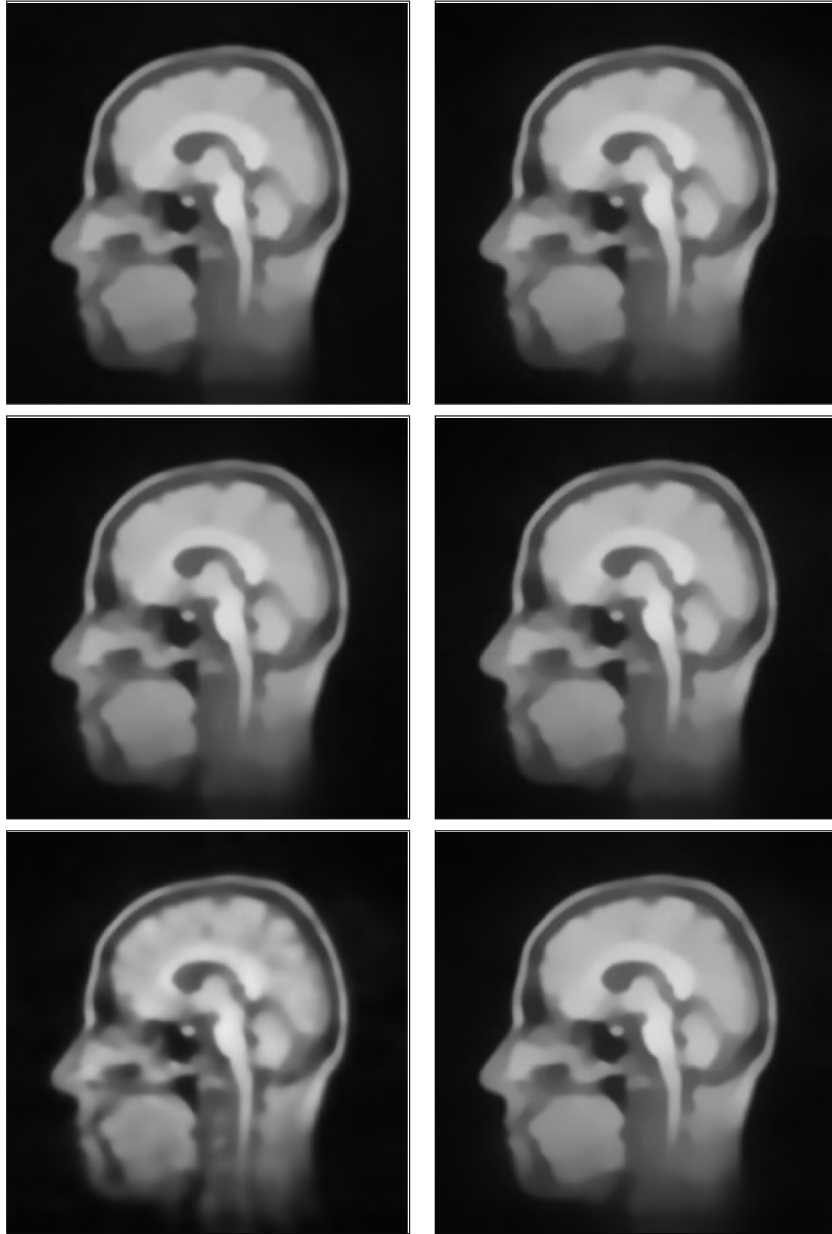


Fig. 5. Comparing deterministic and minimally stochastic computations of TV diffusion filtering with total diffusion time 100.

Left column: Deterministic calculation with explicit scheme.

Right column: Minimally stochastic calculation.

From top to bottom: Time step size $\tau = 0.01$, 0.1 , and 1 requiring 10^4 , 10^3 , and 10^2 iterations, respectively.

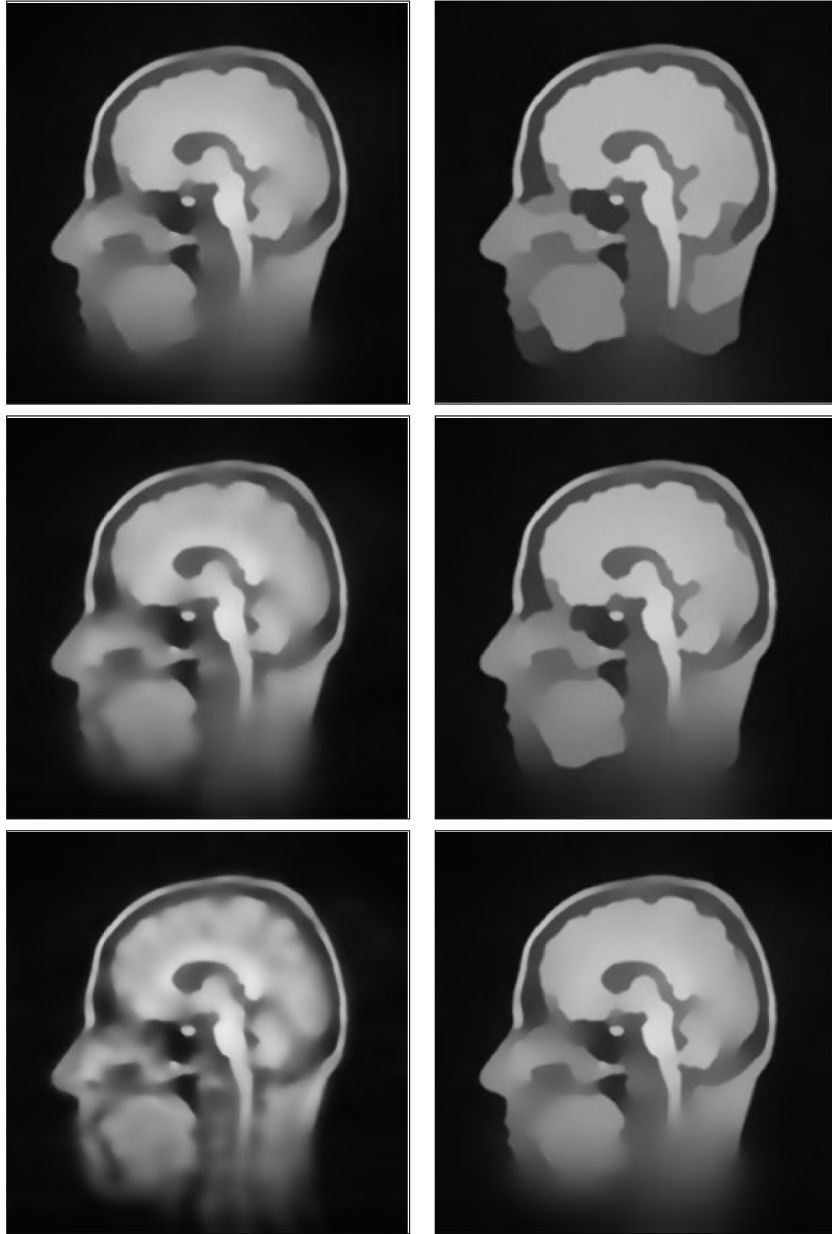


Fig. 6. Comparing deterministic and minimally stochastic computations of BFB diffusion filtering with total diffusion time 3000.

Left column: Deterministic calculation with explicit scheme.

Right column: Minimally stochastic calculation.

From top to bottom: Time step size $\tau = 3, 10$, and 30 requiring 1000, 300, and 100 iterations, respectively.

is required. The numerical experiments show the favourable performance of the minimally stochastic scheme.

Ongoing research dedicated to the general class of diffusivities $g(|\nabla u|) = \frac{1}{|\nabla u|^p}$, $p > 0$, encompasses the usage of a more sophisticated four-pixel scheme and a deeper investigation of the performance.

References

1. L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations in image processing. *Arch. Ration. Mech. Anal.*, 123:199–257, 1993.
2. F. Andreu, C. Ballester, V. Caselles, and J. M. Mazón. Minimizing total variation flow. *Differ. Integr. Equat.*, 14(3):321–360, March 2001.
3. F. Andreu, V. Caselles, J. I. Diaz, and J. M. Mazón. Qualitative properties of the total variation flow. *J. Funct. Anal.*, 188(2):516–547, February 2002.
4. G. Bellettini, V. Caselles, and M. Novaga. The total variation flow in R^N . *J. Differ. Equat.*, 184(2):475–525, 2002.
5. T. Brox, M. Welk, G. Steidl, and J. Weickert. Equivalence results for TV diffusion and TV regularisation. In L. D. Griffin and M. Lillholm, editors, *Scale-Space Methods in Computer Vision*, volume 2695 of *Lecture Notes in Computer Science*, pages 86–100, Berlin, 2003. Springer.
6. F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.*, 32:1895–1909, 1992.
7. Q. S. Chang and I. Chern. Acceleration methods for total variation based denoising problems. *SIAM J. Sci. Comput.*, 25:982–994, 2003.
8. F. Dibos and G. Koepfler. Global total variation minimization. *SIAM J. Numer. Anal.*, 37(2):646–664, 2000.
9. F. Dibos and G. Koepfler. Total variation minimization by the Fast Level Sets Transform. In *Proc. First IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 145–152, Vancouver, Canada, July 2001. IEEE Computer Society Press.
10. S. L. Keeling and R. Stollberger. Nonlinear anisotropic diffusion filters for wide range edge sharpening. *Inverse Probl.*, 18:175–190, January 2002.
11. P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12:629–639, 1990.
12. I. Pollak, A. S. Willsky, and Y. Huang. Nonlinear evolution equations as fast and exact solvers of estimation problems. *IEEE Trans. Signal Process.*, 2004.
13. U. S. Ranjan and K. R. Ramakrishnan. A stochastic scale space for multiscale image representation. In M. Nielsen, P. Johansen, O. F. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 441–446. Springer, Berlin, 1999.
14. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
15. G. Steidl and J. Weickert. Relations between soft wavelet shrinkage and total variation denoising. In L. Van Gool, editor, *Pattern Recognition*, volume 2449 of *Lecture Notes in Computer Science*, pages 198–205. Springer, Berlin, 2002.
16. V. I. Tsurkov. An analytical model of edge protection under noise suppression by anisotropic diffusion. *J. Comput. Syst. Sci. Int.*, 39(3):437–440, 2000.

17. J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. Image Process.*, 7(3):398–410, March 1998.
18. M. Welk, J. Weickert, and G. Steidl. A four-pixel scheme for singular differential equations. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale-Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 585–597, Berlin, 2005. Springer.

Part V

Image Registration

Total Variation Based Image Registration

Claudia Frohn-Schauf^{*1}, Stefan Henn¹, Lars Hömke^{1,2}, and Kristian Witsch¹

¹ Mathematisches Institut, Heinrich-Heine-Universität Düsseldorf,
Universitätsstraße 1, D-40225 Düsseldorf, Germany. E-mail:
{frohn,henn,witsch}@am.uni-duesseldorf.de, url:
<http://www.am.uni-duesseldorf.de/~{frohn,henn,witsch}>

² Institut für Medizin, Forschungszentrum Jülich GmbH, D-52425 Jülich,
Germany. E-mail: l.hoemke@fz-juelich.de

Summary. We consider the image registration problem, i.e., to find a reasonable displacement field, such that a transformed template image becomes similar to a so-called reference image. This yields a nonlinear ill-posed inverse problem. The behavior of image registration problems is governed by an energy functional, which measures the disparity between the images. In order to minimize the matching energy, we replace it by a quadratic approximation. The resulting quadratic minimization problem is also ill-posed, since the associated Hessian is ill-conditioned.

A common approach is to replace the Hessian by the so-called Navier-Lamé operator from linear elasticity theory. This regularization results in a globally smooth displacement field and becomes poor if discontinuities are present in the actual displacement field. Therefore we propose a total variation based regularization that improves the condition of the problem while not penalizing discontinuities in the displacement field. Finally, numerical experiments demonstrate the capabilities of the proposed approach.

Key words: Total variation, image registration, regularization, nonlinear systems, augmented Lagrangian

1 Introduction.

In this paper we consider the image registration problem, i.e., to find a reasonable displacement field $u(x) = (u_1(x), u_2(x))^T$, such that a transformed template image $T(x - u(x))$ becomes similar to a so-called reference image $R(x)$. Image registration, also known as image matching or image mapping, is a widely-used method in medical image analysis, having applications in various domains, see e.g. [3, 6, 16, 20, 25, 45]. A good survey of a part of the

^{*}The work of this author was supported by the Deutsche Forschungsgemeinschaft; grant HE 3404.

practical applications is given in [7, 32] and the references therein. There is a rich theory and also a large number of algorithms to solve the image registration problem, a good survey is given in [17, 33]. They all ask for an ‘optimal’ transformation, which deforms one image such that there is an ‘optimal’ correlation to another image with respect to a suitable coherence or difference measure $D(T, R; u)$.

Numerical optimization is required to minimize the functional $D(T, R; u)$ over a function space $\mathcal{X}(\Omega)$. For instance, in the situation that the intensities of the given images are comparable, a common approach is to minimize the squared differences

$$D_{SSD}(T, R; u(x, y)) = \int_{\Omega} (T(x - u_1(x, y), y - u_2(x, y)) - R(x, y))^2 dx dy. \quad (1)$$

It is used, for example, in the case that the images are recorded with the same imaging machinery, the so-called mono-modal image registration.

One classical approach to determine the displacement fields is the optical flow computation (OFC). Recently, a lot of good ideas have been developed for practical and theoretical studies on OFC for an image sequence $I(x, t)$, see e.g. [4, 2, 30, 47, 28, 10]. Almost all these approaches use the classical ‘brightness constancy assumption’, i.e., the linearization

$$I(x - u(x, t), t + 1) = I(x, t) - \nabla I(x, t) \cdot u(x, t) \quad (2)$$

is assumed to be exact. By using $v(x, t) = \frac{u(x, t)}{\partial t}$ and

$$I_t(x, t) = \frac{I(x - u(x, t), t + 1) - I(x, t)}{\partial t}$$

equation (2) can be transformed to the so-called optical flow equation:

$$\nabla I(x, t) \cdot v(x, t) + I_t(x, t) = 0.$$

The most classical approaches in OFC rule out discontinuous and irregular solutions by adding a regularizing term to an attached term on the data and search for a solution among the minima of an energy term. The most common approach proposed by Horn and Schunk [29] imposes global smoothness constraints for the velocity field on the solution of the optical flow equation. It has been shown by Christoph Schnörr [41] that the resulting functional has a unique minimizer (close to the identity, i.e., $v(x, t) = 0$) in $H^1(\Omega) \times H^1(\Omega)$ and depends continuously on the given image data. Another approach is used by Nagel and Enkelmann [34], where the smoothing depends on the intensity of the template image and not on the displacements itself. A survey of the ‘state of the art’ in OFC is given in [8, 5] and the references therein.

Another kind of problem is the so-called multimodality image matching (see e.g. [48, 31, 27, 19, 26]). Here, the distance between the images is measured by mutual information or entropy based functionals.

Marc Droske and Martin Rumpf [22] have presented an approach based on the definition of a matching energy, which measures the local ‘morphological defect’ between the images.

Here, we present a minimization scheme for the image registration problem, that works for an arbitrary matching energy D . Therefore, we consider the linearization

$$D(u^{(k)} + v^{(k)}) \approx D(u^{(k)}) + \left(\nabla D(u^{(k)}), v^{(k)} \right) + \frac{1}{2} \left(H_D(u^{(k)}) v^{(k)}, v^{(k)} \right)$$

of D around a current approximation $u^{(k)}$ of the displacement field. The resulting quadratic minimization problem is ill-posed, due to the fact that determining the unknown components of the displacements merely from the images is an underdetermined problem, see [24]. Consequently, regularization techniques have to be applied in order to compute meaningful solutions. Furthermore, regularization techniques incorporate desired features of the displacements into the minimization problem and they determine what part of $\mathcal{X}(\Omega)$ is preserved and what part is eliminated. For instance, the most classical regularization term is related to the mechanical stress of a stretched elastic material (see e.g. [16, 24, 25, 27]). Here, the minimization over $H^1(\Omega) \times H^1(\Omega) \subset \mathcal{X}(\Omega)$ may be interpreted physically as the deformation of an elastic membrane and results in a globally smooth displacement field. This approach becomes poor if discontinuities in the displacement field (resulting e.g. from multiple moving objects or partially occluded objects) are expected.

To overcome this, in this paper we propose a total variation based regularization. The advantage of the total variation regularization is that it does not penalize smooth transitions in the displacement field while it recovers discontinuities in the displacement field. The issue of total variation minimization can be traced back to the classical paper of Rudin, Osher and Fatemi [39]. In recent years there has been an increasing number of papers devoted to the use of total variation regularization in image processing. The fundamental importance of this approach can be seen from the following, probably incomplete, list of applications: denoising, deblurring, blind deconvolution, inpainting and optical flow computation (OFC), see [4, 13, 11, 15, 12, 14, 21, 36, 39, 46].

This paper is organized as follows. In Section 2.1, the BV seminorm is described for vector valued functions, i.e., the displacement fields. In Section 2.2 we present a minimization strategy for the image registration problem on the basis of total variation regularization. Then, in Section 3.1., a suitable discretization and linearization of the underlying nonlinear partial differential equation (PDE) is given. To solve this problem numerically, we use the Sequential Quadratic Programming (SQP) framework, see [35]. In Section 4 we present a real image registration example that demonstrates the capabilities of the total variation approach considered. The last section is devoted to some concluding remarks.

2 Continuous Total Variation Minimization.

In this section we propose a novel regularization energy for the image registration problem based on the seminorm of the space of the functions of bounded variation BV on Ω .

2.1 Bounded Variation Seminorm

The space of functions of bounded variation on a domain Ω with Lipschitz continuous boundary is given by

$$BV(\Omega) = \{u \in L^1(\Omega); \quad \mathcal{TV}(u) < \infty\}$$

with total variation

$$\mathcal{TV}(u) = \int_{\Omega} |\nabla u(x, y)| \, dx dy. \quad (3)$$

We get

$$\mathcal{TV}(u) = \int_{\Omega} \left(\sqrt{u_x^2} + \sqrt{u_y^2} \right) \, dx dy$$

by using the 1-norm for the gradient of u in (3), see [23] for a comparison between the 1- and the 2-norm for the image denoising problem. The total variation is the BV seminorm, since the null space of $\mathcal{TV}(u)$ consists of the constant functions. For a vector function $\mathbf{u} = (u_1, u_2) \in BV(\Omega) \times BV(\Omega)$, we define the total variation as follows:

$$\mathcal{TV}(\mathbf{u}) := \mathcal{TV}(u_1) + \mathcal{TV}(u_2). \quad (4)$$

In order to handle the singularities in flat regions where $|\nabla u| \approx 0$, we use a common method (see e.g. [1]) which obtains a regularization by replacing the total variation $\mathcal{TV}(u)$ by a smooth approximation

$$\mathcal{TV}_{\beta}(u) = \int_{\Omega} \left(\sqrt{u_x^2 + \beta} + \sqrt{u_y^2 + \beta} \right) \, dx dy$$

and $\mathcal{TV}(\mathbf{u})$ in (4) by

$$\mathcal{TV}_{\beta}(\mathbf{u}) = \mathcal{TV}_{\beta}(u_1) + \mathcal{TV}_{\beta}(u_2).$$

Taking the formal first variation of $\mathcal{TV}_{\beta}(u)$ leads to

$$\frac{d}{d\varepsilon} \mathcal{TV}_{\beta}(u + \varepsilon\varphi) \Big|_{\varepsilon=0} = \int_{\Omega} \left(\frac{u_x \varphi_x}{\sqrt{u_x^2 + \beta}} + \frac{u_y \varphi_y}{\sqrt{u_y^2 + \beta}} \right) \, dx dy$$

for $\varphi \in BV(\Omega)$. Integration by parts leads to

$$\begin{aligned} \int_{\Omega} \left(\frac{u_x \varphi_x}{\sqrt{u_x^2 + \beta}} + \frac{u_y \varphi_y}{\sqrt{u_y^2 + \beta}} \right) dx dy &= - \int_{\Omega} \left(\left(\frac{u_x}{\sqrt{u_x^2 + \beta}} \right)_x \varphi + \left(\frac{u_y}{\sqrt{u_y^2 + \beta}} \right)_y \varphi \right) dx dy \\ &\quad + \int_{\partial\Omega} \left(\frac{u_x}{\sqrt{u_x^2 + \beta}} n_1 \varphi + \frac{u_y}{\sqrt{u_y^2 + \beta}} n_2 \varphi \right) ds, \end{aligned}$$

where $\mathbf{n} = (n_1, n_2)$ refers to the outward unit normal to the boundary $\partial\Omega$ of Ω . Hence, we can conveniently define the following nonlinear diffusion operator

$$\mathcal{L}(u(x, y))u(x, y) := -\operatorname{div} \left((\kappa_{\beta}(u(x, y)), \nabla u(x, y)) \right) \quad (5)$$

with

$$\kappa_{\beta}(u) : u \longrightarrow \left(\begin{array}{c} \frac{1}{\sqrt{u_x^2 + \beta}} \\ \frac{1}{\sqrt{u_y^2 + \beta}} \end{array} \right).$$

2.2 A $\mathcal{TV}$ Based Minimization Strategy for Image Registration

Consider the quadratic approximation

$$D(T, R; \mathbf{u}^{(k+1)}) \approx D(T, R; \mathbf{u}^{(k)}) + D_u(\mathbf{v}^{(k)}) + \frac{1}{2}(\mathcal{H}_D(\mathbf{u}^{(k)})\mathbf{v}^{(k)}, \mathbf{v}^{(k)})$$

of the matching energy $D(T, R; \mathbf{u}^{(k+1)})$, with the first Gâteaux derivative $D_u(\mathbf{v}^{(k)})$ of D in direction $\mathbf{v}^{(k)}(x, y) = (v_1^{(k)}(x, y), v_2^{(k)}(x, y))$ defined by

$$\begin{aligned} D_u(\mathbf{v}^{(k)}) &= \lim_{s \rightarrow 0} \frac{D(\mathbf{u}^{(k)}(x, y) + s\mathbf{v}^{(k)}(x, y)) - D(\mathbf{u}^{(k)}(x, y))}{s} \\ &= \left(\mathcal{J}_D(\mathbf{u}^{(k)}(x, y)), \mathbf{v}^{(k)}(x, y) \right), \end{aligned}$$

the Jacobian

$$\mathcal{J}_D(\mathbf{u}^{(k)}(x, y)) =: -f(\mathbf{u}^{(k)}(x, y))$$

with steepest descent direction

$$f(\mathbf{u}^{(k)}) = \left(f_1(\mathbf{u}^{(k)}), f_2(\mathbf{u}^{(k)}) \right)^t,$$

and the Hessian $\mathcal{H}_D(\mathbf{u}^{(k)})$ of D at $\mathbf{u}^{(k)}$. For a given current approximation $\mathbf{u}^{(k)}$ we search for a descent direction $\mathbf{v}^{(k)}$ in each iteration step, so that

$$D(T, R; \mathbf{u}^{(k+1)}) < D(T, R; \mathbf{u}^{(k)}).$$

with

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{v}^{(k)}.$$

Since image registration is generally an ill-posed problem, see [25], neither the original minimization problem nor the quadratic minimization problem

$$\min \mathcal{Q}(\mathbf{v}^{(k)}) = \min \left\{ D(T, R; \mathbf{u}^{(k)}) + (\mathcal{J}_D(\mathbf{u}^{(k)}), \mathbf{v}^{(k)}) + \frac{1}{2}(\mathcal{H}_D(\mathbf{u}^{(k)})\mathbf{v}^{(k)}, \mathbf{v}^{(k)}) \right\},$$

need to have a solution. Thus, regularization techniques have to be applied in order to compute meaningful solutions. To overcome this instability we consider the following regularized minimization problem

$$\min \tilde{\mathcal{Q}}(\mathbf{v}^{(k)}) = \min \left\{ D(\mathbf{u}^{(k)}) + (\mathcal{J}_D(\mathbf{u}^{(k)}), \mathbf{v}^{(k)}) + \mathcal{TV}_\beta(\mathbf{v}^{(k)}) \right\}, \quad (6)$$

where the bilinear form $(\mathcal{H}_D(\mathbf{u}^{(k)})\mathbf{v}^{(k)}, \mathbf{v}^{(k)})$ with ill-conditioned Hessian is replaced by the bounded variation seminorm. There are two important aspects to this choice:

- the Hessian of $\mathcal{TV}_\beta(\mathbf{v}^{(k)})$ is symmetric and positive semidefinite,
- it permits discontinuities in the solution $\mathbf{v}^{(k)}$, but also enforces regularity of the displacement field.

For $H^1(\Omega) \times H^1(\Omega)$ regularization methods, e.g. elastic regularization, the latter is not the case.

According to the fundamental lemma of the calculus of variations (cf. [18]) the formal first variation (Euler-Lagrange equation) of (6) using (5) is given by

$$(\mathcal{J}_D(\mathbf{u}^{(k)}) + \mathcal{L}(\mathbf{v}^{(k)})\mathbf{v}^{(k)}, \varphi) = 0. \quad (7)$$

Consequently, a minimizer of (6) is a weak solution of the following nonlinear boundary value problem

$$\begin{aligned} -\operatorname{div} \left((\kappa_\beta(v_1^{(k)}(x, y)), \nabla v_1^{(k)}(x, y)) \right) &= f_1^{(k)}(x, y) \\ -\operatorname{div} \left((\kappa_\beta(v_2^{(k)}(x, y)), \nabla v_2^{(k)}(x, y)) \right) &= f_2^{(k)}(x, y) \end{aligned}$$

supplemented by the Neumann boundary conditions

$$\frac{\partial v_1^{(k)}(x, y)}{\partial n} = \frac{\partial v_2^{(k)}(x, y)}{\partial n} = 0 \quad \text{for all } (x, y) \in \partial\Omega.$$

The PDE can be expressed in operator form

$$\mathcal{L}(\mathbf{v}^{(k)}(x, y))\mathbf{v}^{(k)}(x, y) = f^{(k)}(x, y). \quad (8)$$

The nonlinear system (8) is singular, since the operator $\mathcal{L}$ has a non-trivial kernel. The operator $\mathcal{L}$ is symmetric, therefore by the well known Fredholm alternative it follows, that equation (8) is solvable if

$$f^{(k)}(x, y) \perp \ker(\mathcal{L}) \Leftrightarrow \int_{\Omega} f_1^{(k)}(x, y) dx dy + \int_{\Omega} f_2^{(k)}(x, y) dx dy = 0 \quad (9)$$

and the solution is unique if

$$\mathbf{v}^{(k)} \perp \ker(\mathcal{L}) \Leftrightarrow \int_{\Omega} v_1^{(k)}(x, y) dx dy + \int_{\Omega} v_2^{(k)}(x, y) dx dy = 0. \quad (10)$$

3 Numerical Minimization

3.1 Finite Difference Discretization

Images are typically encoded as two-dimensional arrays. Each element in the matrix represents a pixel (picture element) with gray intensity between black and white (0 and 255).

The resulting image array is a finite-dimensional approximation of a continuous image and is represented on a rectangular equidistant grid

$$g_{i,j}^h = (x_i, y_j) = (ih_x, jh_y) \in \bar{\Omega}, \quad 0 \leq i \leq n_x, \quad 0 \leq j \leq n_y$$

with $(n_x + 1) \times (n_y + 1)$ grid points with pixel-wide h_x and pixel-height h_y given by

$$h = (h_x, h_y) = (n_x^{-1}, n_y^{-1}).$$

Let $u_l^h(i, j)$, $f_l^h(i, j)$ and $|\nabla u_l^h(i, j)|$ denote the grid functions defined by $u_l^h(i, j) = u_l(g_{i,j}^h)$, $f_l^h(i, j) = f_l(g_{i,j}^h)$ and $|\nabla u_l^h(i, j)| = |\nabla u_l(g_{i,j}^h)|$ as well as their second-order finite difference approximations at staggered grid points

$$D_x^{\frac{h_x}{2}} u_l^h(i + 1/2, j) = (u_l^h(i + 1, j) - u_l^h(i, j))/h_x$$

and

$$D_y^{\frac{h_y}{2}} u_l^h(i, j + 1/2) = (u_l^h(i, j + 1) - u_l^h(i, j))/h_y.$$

For simplicity, we will drop the dependence on h from the notations in the following. We can now replace the spatial derivatives in the operator $\mathcal{L}$ by their second-order finite difference approximations. The discretized differential operator $\mathcal{L}_h$ is given by

$$\mathcal{L}_h = \left[\begin{array}{ccc|ccc} 0 & a_1^y(i, j + \frac{1}{2}) & 0 & 0 & 0 & 0 \\ a_1^x(i - \frac{1}{2}, j) & \Sigma_1(i, j) & a_1^x(i + \frac{1}{2}, j) & 0 & 0 & 0 \\ 0 & a_1^y(i, j - \frac{1}{2}) & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & a_2^y(i, j + \frac{1}{2}) & 0 \\ 0 & 0 & 0 & a_2^x(i - \frac{1}{2}, j) & \Sigma_2(i, j) & a_2^x(i + \frac{1}{2}, j) \\ 0 & 0 & 0 & 0 & a_2^y(i, j - \frac{1}{2}) & 0 \end{array} \right]$$

with

$$a_l^x(i, j) = -\frac{h_x^{-2}}{\sqrt{(D_x^{\frac{h_x}{2}} u_l^h(i, j))^2 + \beta}}, \quad a_l^y(i, j) = -\frac{h_y^{-2}}{\sqrt{(D_y^{\frac{h_y}{2}} u_l^h(i, j))^2 + \beta}} \quad (11)$$

and

$$\Sigma_l(i, j) = -\left(a_l^x(i - \frac{1}{2}, j) + a_l^x(i + \frac{1}{2}, j) + a_l^y(i, j - \frac{1}{2}) + a_l^y(i, j + \frac{1}{2}) \right)$$

for $l = 1, 2$. First order approximations are used for the outward normal of the boundary value problem. The proposed discrete difference discretization of the

boundary value problem leads to $m = 2 \cdot n$ equations with $n = (n_x + 1) \cdot (n_y + 1)$ and can be written as

$$A(\mathbf{u}^{(k)}) \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}$$

where

$$\mathbf{f}_1 = \text{vec}(f_1^h) = (f_1^h(0, 0), \dots, f_1^h(n_x, n_y))^T \in \mathbb{R}^n$$

the discretized function vector of all picture elements (in a component-wise ordering) with the map $\text{vec} : \mathbb{R}^{(n_x+1) \times (n_y+1)} \rightarrow \mathbb{R}^{(n_x+1) \cdot (n_y+1)}$. The so-called discretization matrix $A \in \mathbb{R}^{m \times m}$ is symmetric and positive definite, but not regular. Let

$$\mathbb{I}_n := \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n,$$

then the kernel of A is spanned by the vectors

$$b_1 = \begin{bmatrix} \mathbb{I}_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^m \quad \text{and} \quad b_2 = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathbb{I}_n \end{bmatrix} \in \mathbb{R}^m.$$

We define the matrix B by

$$B := \begin{bmatrix} b_1 & b_2 \end{bmatrix} \in \mathbb{R}^{m \times 2},$$

with $AB = A^t B = \mathbf{0}$.

3.2 Sequential Quadratic Programming (SQP) Framework

To deal with the nonlinearity of the total variation $\mathcal{TV}_\beta$ in equation (6) we propose to solve the quadratic optimization problem

$$\text{Minimize}_{\mathbf{v}^{(k)}} \quad (\mathcal{J}_D(\mathbf{u}^{(k)}), \mathbf{v}^{(k)}) + \frac{1}{2} (A(\mathbf{v}^{(k-1)}) \mathbf{v}^{(k)}, \mathbf{v}^{(k)})$$

subject to

$$(B, \mathcal{J}_D(\mathbf{u}^{(k)})) = - \sum_{i=1}^2 (\mathbb{I}_n, \mathbf{f}_i^{(k)}) = 0 \quad (12)$$

and

$$(B, \mathbf{v}^{(k)}) = \sum_{i=1}^2 (\mathbb{I}_n, \mathbf{v}_i^{(k)}) = 0, \quad (13)$$

instead. This quadratic minimization problem is numerically more tractable than equation (6). We only have to solve the linear PDE

$$\mathcal{L}(\mathbf{v}^{(k-1)}(x, y))\mathbf{v}^{(k)}(x, y) = f(\mathbf{u}^{(k)}(x, y)) \quad (14)$$

in each iteration step, subject to the side conditions (12) and (13). The first side condition can be enforced by the modified gradient $\mathcal{J}_D^{(k)}$ of D :

$$\mathcal{J}_D^{(k)} = (I - BB^+) \mathcal{J}_D(\mathbf{u}^{(k)})$$

where $P = I - BB^+$ is an orthogonal projector onto the range of $A(\mathbf{v}^{(k-1)})$. The side condition (13) enforces the orthogonality condition $\mathbf{v}^{(k)} \perp \mathbb{I}$. By using Lagrange multipliers $\lambda^{(k)} = (\lambda_1^{(k)}, \lambda_2^{(k)})^t \in \mathbb{R}^{2 \times 1}$, the minimization of D consists of a sequence of unconstrained minimization subproblems:

$$\min_{\mathbf{v}^{(k)}, \lambda^{(k)}} \Phi(\mathbf{v}^{(k)}, \lambda^{(k)}), \quad (15)$$

where

$$\Phi(\mathbf{v}^{(k)}, \lambda^{(k)}) = (\mathcal{J}_D^{(k)}, \mathbf{v}^{(k)}) + \frac{1}{2} (A(\mathbf{v}^{(k-1)})\mathbf{v}^{(k)}, \mathbf{v}^{(k)}) + (B\lambda^{(k)}, \mathbf{v}^{(k)}).$$

The stationary points of Φ satisfy

$$\nabla \Phi(\mathbf{v}^{(k)}, \lambda^{(k)}) = \begin{pmatrix} \mathcal{J}_D^{(k)} + A(\mathbf{v}^{(k-1)})\mathbf{v}^{(k)} + B\lambda^{(k)} \\ B^T \mathbf{v}^{(k)} \end{pmatrix} = 0.$$

Consequently, at each Newton step, we are faced with

- 1) *solving* the following symmetric linear so-called Karush–Kuhn–Tucker (KKT) system of equations

$$\begin{bmatrix} A(\mathbf{v}^{(k-1)}) & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}^{(k)} \\ \lambda^{(k)} \end{bmatrix} = \begin{bmatrix} -\mathcal{J}_D^{(k)} \\ 0 \end{bmatrix} \quad (16)$$

- 2) *update* $\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \alpha^{(k)} \mathbf{v}^{(k)}$ with a linesearch parameter $\alpha^{(k)}$ given by the following one-dimensional minimization problem:

$$\alpha^{(k)} = \operatorname{argmin}_{\alpha > 0} D(\mathbf{u}^{(k)} + \alpha \mathbf{v}^{(k)}),$$

- 3) *check* $D(\mathbf{u}^{(k+1)}) < D(\mathbf{u}^{(k)})$.

3.3 Numerical Solution of the Resulting KKT System

The focus of this paper is not on fast numerical solvers for the linear KKT system (16), which is the time-consuming part of the minimization process. Solving KKT systems is a well known task and has been addressed by many authors, see [9] for an overview. For convenience we solve the system by using a direct solver in this introductory work. However, particularly for the treatment of the nonlinear case the favored solution procedure involves nonlinear full approximation scheme (FAS) multigrid techniques as well as multiscale preconditioning schemes as described in [23] for the image denoising problem.

4 Results

In this section numerical experiments for a synthetic and a human brain example are presented in order to demonstrate the qualities of the proposed approach. For both examples, we use the least squares disparity measure (1) to measure the distance between template and reference.

4.1 A Synthetic Example

We start with a synthetic example of size 128×128 . Both template (Figure 1(a)) and the reference (Figure 1(b)) contain two rectangles, a white rectangle and a gray rectangle that is stacked on top of the white rectangle. The exact displacement field is given by a shift of the upper rectangle to the right and a shift of the lower rectangle to the left. Consequently the exact displacement field is piecewise constant with significant discontinuities between the upper and the lower rectangle. Thus, the registration process using an isotropic $H^1(\Omega) \times H^1(\Omega)$ registration must fail.

The result obtained with the proposed total variation based minimization approach is shown in Figure 1(c). Both rectangles in the template image are matched onto the corresponding rectangles in the reference image. We now compare this result with the one obtained with an elastic regularization based minimization approach. In each iteration step the linear system

$$\begin{bmatrix} E_h(\mu, \lambda) & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}^{(k)} \\ \lambda^{(k)} \end{bmatrix} = \begin{bmatrix} \mathcal{J}_D^{(k)} \\ 0 \end{bmatrix}$$

has to be solved. $E_h(\mu, \lambda)$ is the discretization of the so-called Navier-Lamé operator

$$\begin{aligned} E(\mu, \lambda) \mathbf{v}(x, y) &= -\mu \Delta \mathbf{v}(x, y) - (\lambda + \mu) \nabla \operatorname{div} \mathbf{v}(x, y) \\ &= \begin{cases} -\mu \Delta v_1(x, y) - (\lambda + \mu) \left(\frac{\partial^2 v_1(x, y)}{\partial x^2} + \frac{\partial^2 v_2(x, y)}{\partial x \partial y} \right) \\ -\mu \Delta v_2(x, y) - (\lambda + \mu) \left(\frac{\partial^2 v_1(x, y)}{\partial x \partial y} + \frac{\partial^2 v_2(x, y)}{\partial y^2} \right). \end{cases} \end{aligned}$$

The Lamé constants $\lambda \geq 0$ and $\mu > 0$ reflect the material properties, i.e., the lateral shrink

$$\nu = \frac{\lambda}{2(\lambda + \mu)},$$

and the modulus of elasticity (Young's modulus)

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu}.$$

The Figures 1(d)–(f) display results for elastic regularization for different choices of λ and μ . Obviously, there is a misalignment in the region where the rectangles are in direct contact. The 64th column of the first component of the

displacement fields for both regularizers is displayed in Figure 2. Total variation regularization yields a discontinuous transition between the upper and the lower rectangle. In contrast, elastic regularization yields a smooth transition at the discontinuity, because discontinuities are penalized. The computed displacement fields for both regularizers are plotted in Figure 3. The arrows correspond to the components (u_1, u_2) at the image points.

Note that the exact displacement field is symmetric in y -direction, but neither the result based on elastic regularization nor the result for the proposed $\mathcal{TV}$ based regularization is symmetric, see Figure 2 and Figure 3. The reason for this effect is that the right hand side f is based solely on the gray values of the images. In this example the rectangles are represented by different gray values which leads to stronger inner forces for the brighter rectangle and consequently the resulting displacement fields are not symmetric.

4.2 A Human Brain Example

An example from the reconstruction of histological data sets is shown in Figure 4. The reference (Figure 4(b)) is a $20\mu m$ thick histological section of a human brain. The three dimensional reconstruction of the histological data sets is a common task in brain research and is complicated, amongst other things, by nonlinear distortions introduced in the cutting process (see e.g. [43, 42, 44, 38, 37, 40]). In order to guarantee structural equivalence between reference and template image and easier evaluation, we use an artificial displacement field to generate the template image. We moved the left temporal lobe downwards, see Figure 4(a). In the original two-dimensional cutting plane the temporal lobe is not connected to the rest of the brain. Hence, it might be moved independently when the sections are processed.

The resulting displacement fields for both regularizers are displayed in Figure 5. For the elastic regularization we use the parameter setting $(\mu, \lambda) = (1, 1)$. Again the arrows are the components (u_1, u_2) at the image points. Although, the exact displacement field shifts only the left temporal lobe, small displacements exist in other parts of the image. This effect is due to the orthogonality condition (10). Note that these displacements are very small for the fixed object and larger for the background. The object is kept in place by the force term. The result of the total variation regularization yields a discontinuous transition between the transformed structure and the background whereas the displacement resulting from the elastic regularization exhibits a smooth transition at the discontinuity, because discontinuities are penalized. The plot of the 30th column of the second component of the displacement fields in Figure 6 illustrates this point further.

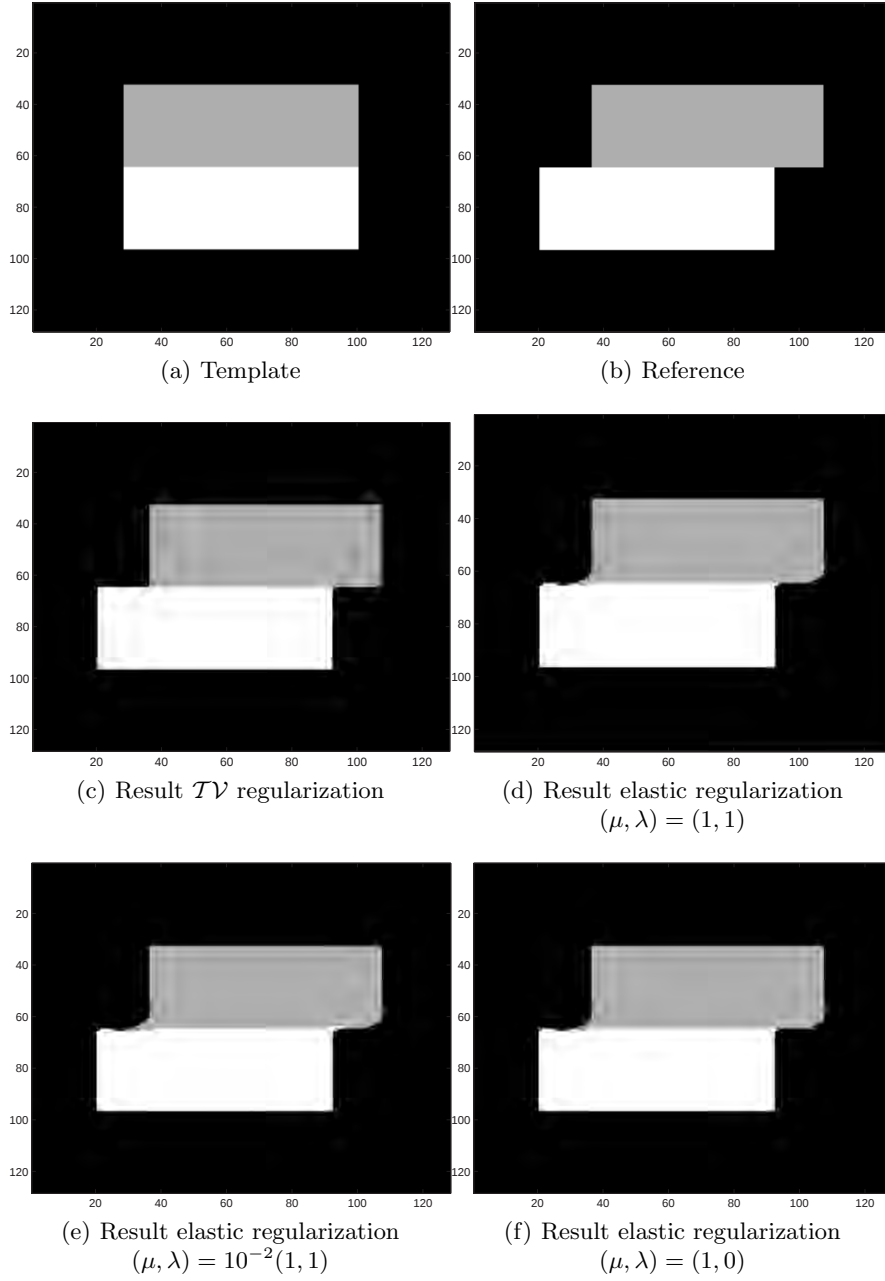


Fig. 1. A synthetic example

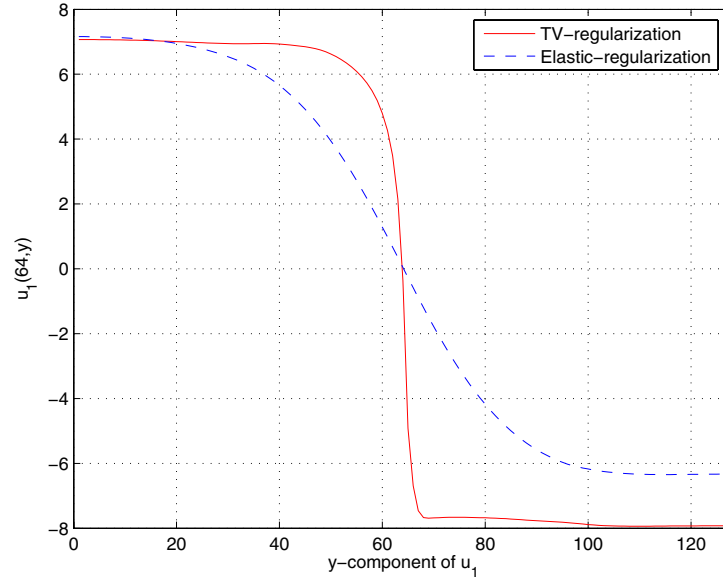


Fig. 2. Plot of the 64th column of $u_1(x, y)$ for the results displayed in Figure 1(c) (TV based regularization) and Figure 1(d) (elastic regularization).

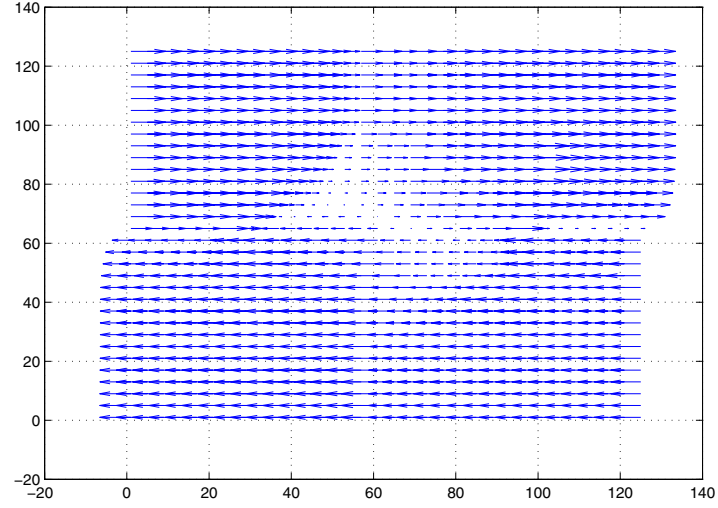
5 Summary and Conclusion

In this paper we present a novel image registration approach based on total variation regularization. This regularization in combination with a SQP method has been shown an attractive method for image registration. The numerical experiments in Section 4 have justified the choice of using total variation for regularizing the displacement fields instead of a conventional isotropic elastic regularization (representative for a $H^1(\Omega) \times H^1(\Omega)$ regularization). It turns out, that the elastic registration fails in the case, that the underlying displacement field contains discontinuities. Our future research along this direction will be mainly targeted on the development of fast multigrid solvers for the KKT system (16).

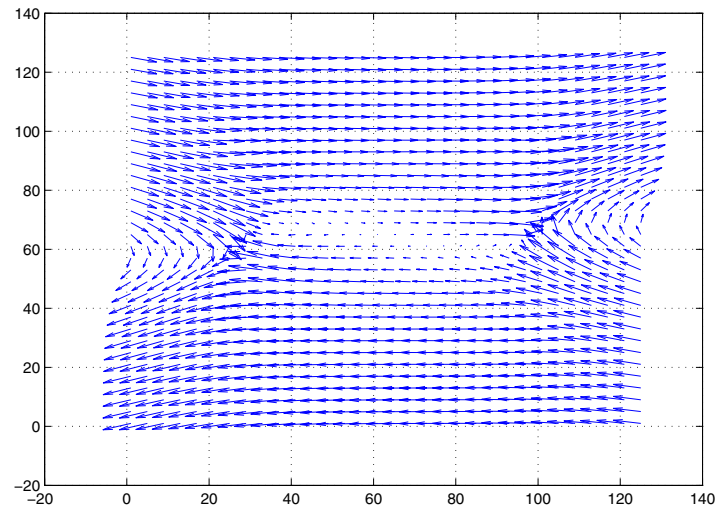
Acknowledgement. Supported by the Deutsche Forschungsgemeinschaft (DFG) (grant HE 3404).

References

1. R. Acar and C.R. Vogel. Analysis of bounded variation penalty method. *Inverse Problems*, 10(6):1217–1229, 1994.



(a) Displacement field using $\mathcal{TV}$ regularization for the result presented in Figure 1(c).



(b) Displacement field using elastic regularization (with parameter setting $(\mu, \lambda) = (1, 1)$) for the result presented in Figure 1(d).

Fig. 3. Displacement fields for the example depicted in Figure 1.

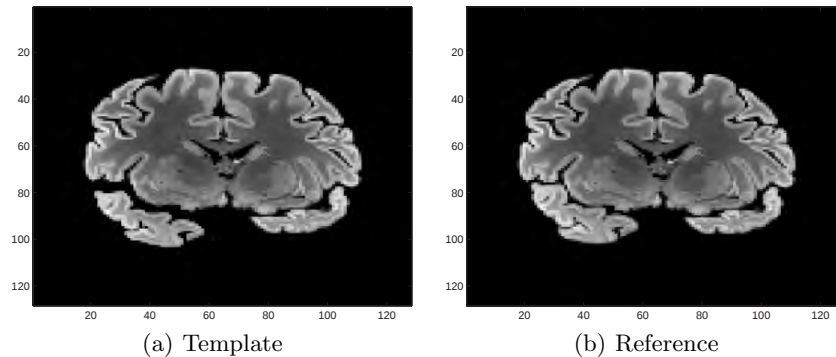
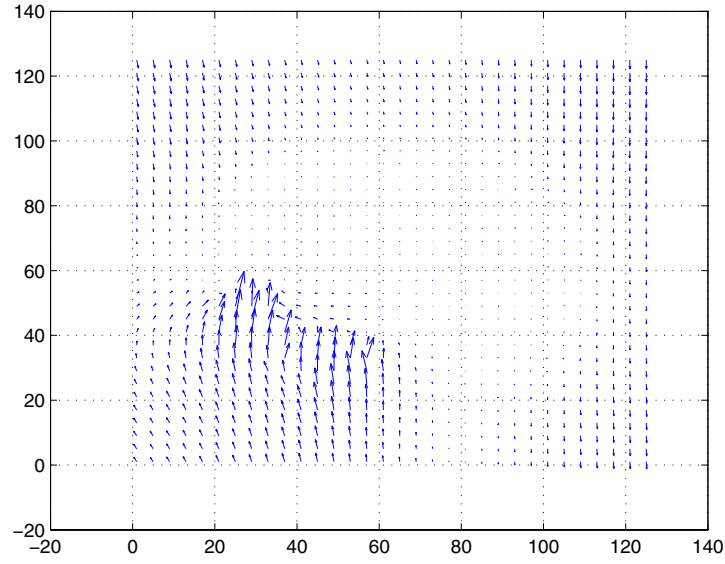
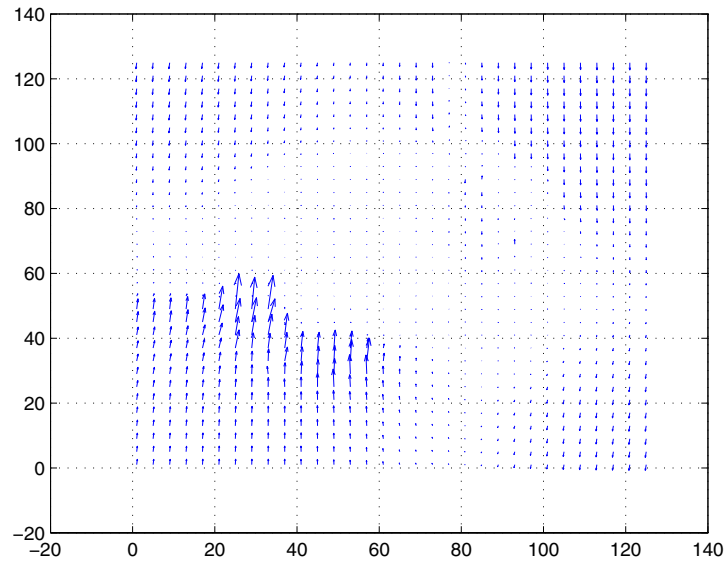


Fig. 4. Human brain example

2. L. Alvarez, J. Weickert, and J. Sánchez. Reliable estimation of dense optical flow fields with large displacements. *Int. J. Comput. Vis.*, 39:41–56, 2000.
3. Y. Amit. A nonlinear variational problem for image matching. *SIAM J. Sci. Comput.*, 15:207–224, 1994.
4. G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. *SIAM J. Appl. Math.*, 60(1):156–182, 1999.
5. G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing – Partial Differential Equations and the Calculus of Variations*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, 2002.
6. R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Computer Vision*, 46:1–21, 1989.
7. A. Barry. Seeking signs of intelligence in the theory of control. *SIAM NEWS*, 30 (3), 1997.
8. S.S. Beauchemin and J.L. Barron. The computation of optical flow. *ACM Comput. Surv.*, 27(3):433–467, 1995.
9. M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numer.*, 14:1–137, 2005.
10. A. Borzi, K. Ito, and K. Kunisch. Optimal control formulation for determining optical flow. *SIAM J. Sci. Comput.*, 24(3):818–847, 2002.
11. A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Numer. Math.*, 76:167–188, 1995.
12. T.F. Chan, R. Chan, and H. Zhou. A continuation method for total variation denoising problems. In *Proceedings of the SPIE Conference on Advanced Signal Processing Algorithms*, 1995.
13. T.F. Chan, G.H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Sci. Comp.*, 20, No. 6:1964–1977, 1999.
14. T.F. Chan, S. Kang, and J. Shen. Euler’s elastica and curvature based inpaintings. *J. Appl. Math.*, 63(2):564–592, 2002.
15. T.F. Chan and K. Wong. Total variation blind deconvolution. *IEEE Trans. Image Proc.*, 7(3):370–375, 1998.
16. G.E. Christensen, M.I. Miller, M. Vannier, and U. Grenander. Individualizing neuroanatomical atlases using a massively parallel computer. *IEEE Computer*, 29(1):32–38, 1996.



(a) Displacement field using elastic regularization, $(\mu, \lambda) = (1, 1)$



(b) Displacement field using TV regularization

Fig. 5. Displacement fields for the human brain example depicted in Figure 4.

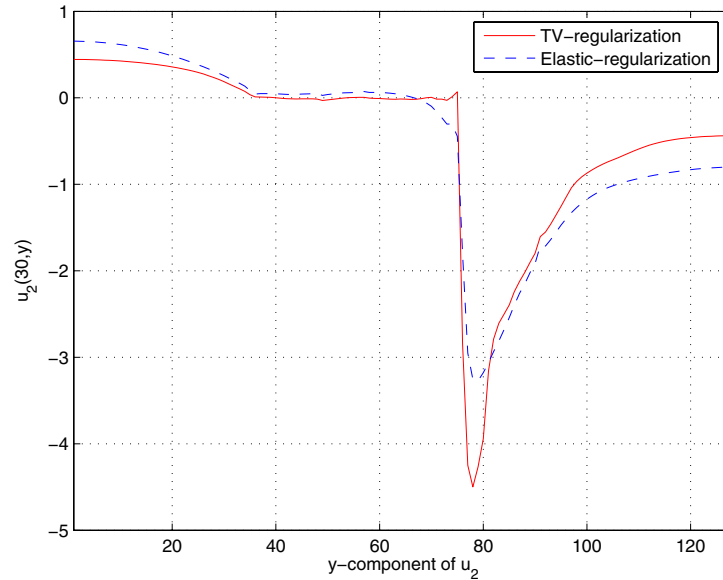


Fig. 6. Plot of the 30th column of $u_2(x, y)$ for the human brain example (Figure 4).

17. U. Clarenz, M. Droske, S. Henn, M. Rumpf, and K. Witsch. Computational methods for nonlinear image registration. In *Otmar Scherzer (ED): Mathematical Method for Registration and Applications to Medical Imaging*, Mathematics in Industry, Vol 10:81–101, 2006.
18. R. Courant. *Calculus of variations*. Courant Institute of Mathematical Sciences, New York, NY, 1962.
19. E. D’Agostino, J. Modersitzki, F. Maes, D. Vandermeulen, B. Fischer, and P. Suetens. Free-form registration using mutual information and curvature regularization. *WBIR*, pages 11–20, 2003.
20. M.H. Davis, A. Khotanzad, D. Flaming, and S. Harms. A physics based coordinate transformation for 3d medical images. *IEEE Trans. Med. Imag.*, 16(3):317–328, 1997.
21. D. C. Dobson and C. R. Vogel. Convergence of an iterative method for total variation denoising. *SIAM J. Numer. Anal.*, 34(5):1779–1791, 1997.
22. M. Droske and M. Rumpf. A variational approach to non-rigid morphological registration. *SIAM Appl. Math.*, 64(2):668–687, 2004.
23. C. Frohn-Schauf, S. Henn, and K. Witsch. Nonlinear multigrid methods for total variation image denoising. *Comput. Visual. Sci.*, 7(3–4):199–206, 2004.
24. S. Henn. A Levenberg-Marquardt scheme for nonlinear image registration. *BIT*, 43(4):743–759, 2003.
25. S. Henn and K. Witsch. Iterative multigrid regularization techniques for image matching. *SIAM J. Sci. Comput.*, 23(4):1077–1093, 2001.
26. S. Henn and K. Witsch. Multi-modal image registration using a variational approach. *SIAM J. Sci. Comput.*, 25(4):1429–1447, 2004.

27. G. Hermosillo. Variational methods for multimodal image matching. *Phd thesis, Université de Nice, France*, 2002.
28. W. Hinterberger, O. Scherzer, C. Schnörr, and J. Weickert. Analysis of optical flow models in the framework of calculus of variations. *Numer. Funct. Anal. Optim.*, 23:69–82, 2002.
29. B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artif. Intell.*, 17:185–203, 1981.
30. M. Lefébure and D. Cohen. Image registration, optical flow, and local rigidity. *JMIV*, 14(2):131–147, 2001.
31. F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Trans. Med. Imag.*, 16(2):187–198, 1997.
32. J. Maintz and M. Viergever. A survey of medical image registration. *Med. Image Anal.*, 2(1):1–36, 1998.
33. J. Modersitzki. *Numerical Methods for Image Registration*. Oxford Science Publications, 2003.
34. H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from images sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(5):565–593, 1986.
35. J. Nocedal and S. J. Wright. *Numerical optimization*. Springer-Verlag, New York, 1999.
36. S. Osher, A. Solé, and L. Vese. Image decomposition and restoration using total variation minimization and the H^1 . *Multiscale Modeling & Simulation*, 1(3):349–370, 2003.
37. S. Ourselin, A. Roche, G. Subsol, X. Pennec, and N. Ayache. Reconstructing a 3D structure from serial histological sections. *Image Vis. Comput.*, 19:25–31, 2000.
38. A. Pitiot, G. Malandain, E. Bardinet, and P. Thompson. Piecewise affine registration of biological images. In J.C. Gee, J.B.A. Maintz, and M.W. Vannier, editors, *Second International Workshop on Biomedical Image Registration WBIR'03*, volume 2717 of *Lecture Notes in Computer Science*, pages 91–101, Philadelphia, PA USA, 1996. Springer-Verlag.
39. L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation-based noise removal algorithms. *Physica D*, 60:259–268, 1992.
40. O. Schmitt and J. Modersitzki. Registrierung einer hochaufgelösten histologischen Schnittserie eines Rattenhirns. In *Bildverarbeitung für die Medizin 2001*, Berlin Heidelberg New York, 2001. Springer-Verlag.
41. Christoph Schnörr. Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class. *Int. J. Comput. Vis.*, 6:25–38, 1991.
42. T. Schormann, A. Dabringhaus, and K. Zilles. Extension of the principal axes theory for the determination of affine transformations. In E. Paulus and F.M. Wahl, editors, *Proceedings of the DAGM: Informatik Aktuell*, pages 384–391, Berlin, 1997. Springer-Verlag.
43. T. Schormann, M. von Matthey, A. Dabringhaus, and K. Zilles. Alignment of 3-D brain data sets originating from MR and histology. *Bioimaging*, 1:119–128, 1993.
44. T. Schormann and K. Zilles. Three-dimensional linear and nonlinear transformations: an integration of light microscopical and MRI data. *Hum. Brain Mapp.*, 6 (5-6):339–347, 1998.

- 45. P. Thompson and A. Toga. Anatomically driven strategies for high-dimensional brain image registration and pathology. *Brain Warping, Academic Press*, pages 311–336, 1998.
- 46. C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM J. Sci. Comp.*, 17(1):227–238, 1996.
- 47. J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *JMIV*, 14(3):245–255, 2001.
- 48. W. Wells, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. *Med. Image Anal.*, 1:35–51, 1996.

Variational Image Registration Allowing for Discontinuities in the Displacement Field

Sven Kabus^{1,2}, Astrid Franz², and Bernd Fischer¹

¹ Institute of Mathematics, University of Lübeck, Wallstr. 40, D-23560 Lübeck, Germany. E-mail: {kabus,fischer}@math.uni-luebeck.de

² Philips Research Laboratories, Röntgenstr. 24–26, D-22335 Hamburg, Germany. E-mail: {sven.kabus, astrid.franz}@philips.com

Summary. Registration of medical images is an active field of current research. The problem is to find a transformation which aligns two given images. The resulting displacement field may be described as a linear combination of pre-selected basis functions (parametric approach), or, as in our case, it may be computed as a minimizer of a functional (non-parametric or variational approach). This functional combines a similarity measure and a smoothness term. The first one puts the comparison of the images into quantifiable terms whereas the latter one regularizes the displacement field. The minimizing task is tackled by computing the Gâteaux derivative of the functional resulting in a set of nonlinear partial differential equations for the displacement field. These equations are linearized by means of a fixed-point iteration scheme and discretized by a standard finite difference approach.

A conventional variational method results in a globally smooth displacement field. However, a variety of clinical applications involve topology changes between the two images as for instance brain shift or tumor appearance or resection. For such applications a generalization of the standard method is needed which allows for localized discontinuities in the displacement field.

The variational image registration approach presented here assumes a segmentation of the images into corresponding subdomains. At the interfaces between neighbouring subdomains the influence of the smoothness term can be suppressed by introducing a spatially dependent weighting function. By choosing it appropriately this allows for opening or closing of a gap between image regions.

We demonstrate the capability of this new registration method by means of a one-dimensional synthetic example and a two-dimensional MR head image. However, our method can be applied to images of arbitrary dimensionality.

Key words: Image registration, finite difference method, variable regularization, discontinuities

1 Introduction

Nonrigid image registration is a challenging field in medical imaging. The task is to find a vector field of displacements such that each point in a template image can be mapped onto a corresponding and meaningful point in the reference image. By the notion ‘meaningful’ often a type of topology preserving constraint is implied. However, there exist several cases where true, physical changes in topology exist and where it is essential to take them into account. For instance, structures which are connected in one image may be separated in the other image, like the brain–skull interface subject to a brain shift. Additionally, structures may move alongside each other, thereby causing discontinuities, like the liver and its surrounding tissue, or, widely spread, a bone structure together with surrounding tissue deforming due to muscle contraction.

Typically, the wanted displacement is computed subject to a smoothness constraint, see, e.g., [1, 9, 2] and references therein. For example, in elastic matching, the constraint is realized by a regularization based on the linear elastic potential of the displacement. Other constraints are based on the curvature of the displacement field or on its gradient, which is the subject of this note. In general, the constraint is applied globally with one global regularization parameter. Usually, the method provides satisfactory results due to the underlying physical model. Nonetheless it fails in cases described above, since a global regularization does not allow for any local changes in the topology.

In the literature one can find several attempts dealing with nonrigid image registration in conjunction with spatially varying regularization or material parameters, for example the radial basis functions [11], the Bezier tensor product [12], the damped springs [4], the finite elements [10, 5, 7] or the finite differences [3] based approaches, respectively. However, these methods either do not reflect the physical behavior of the underlying material, or the registration yields a smooth transformation field, allowing for no discontinuities at all.

Registration tasks with a demand for discontinuities in order to allow for topological changes occur quite often. For purpose of motivation, two three-dimensional MR data sets displaying a patient’s head pre- and intra-operatively are considered. For a tumor resection the skull has been opened causing a brain shift, which can be detected by an increasing gap (consisting of air and liquor) between skull and brain tissue. In the same way this gap is visible after an extraction of one-dimensional line profiles (or 1D images) across the brain–skull interface out of these data sets, as depicted in Figure 1. At a first glance the gap in the intra-operative 1D image (solid line), indicated by nearly zero gray values, does not correspond to any interval in the preoperative 1D image (dashed line). However, when assuming an underlying physical process between the two images the gap interval does correspond to a (very) small interval. Though, due to the discrete character of the images, it may not be visible. Certainly, a registration of this image pair (regardless of which

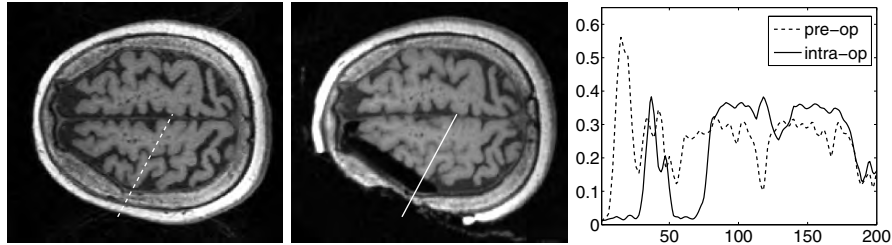


Fig. 1. Slices from 3D MR data sets showing a patient's head pre-operatively (left) and intra-operatively (center) together with extracted line profiles (right).

dimension) is a hard or even unsolvable task for any (non-)parametric approach, which regularizes the deformation in a homogeneous way. Parametric approaches based on, for instance, B-splines or radial basis functions [8, 11], on one side, may employ an adaptive refining. Since changes in topology require a locally highly refined mesh, this yields a complex problem in terms of computational time and storage. On the other side, non-parametric approaches cover highly elastic or even fluidal models. Though a purely fluidal approach [9] does allow for modeling of topological changes, such changes are not restricted to the desired regions like the brain–skull interface for instance and may thus occur everywhere in the image which is not our intent. The use of mixed models based on regions modeled as rigid, elastic or fluidal requires coupling conditions at the interfaces. However, this adds significant complexity to the structure of the equation system to be solved in the numerical part. Consequently, we retain the physical model implied by the problem task and investigate a variable regularization instead.

The note is organized as follows: In Sect. 2 we introduce a widely known variational approach. It is extended in Sect. 3 by a variable regularizer. Sect. 4 reports on preliminary numerical results.

2 Variational Approach

To clarify and investigate the task of registration allowing for discontinuities, in the following subsection we construct a synthetic example related to the extracted 1D images. An application of the standard variational approach reveals its shortcomings in Subsect. 2.3.

2.1 A Synthetic Example

Let $T, R : [a, b] \rightarrow [0, 1]$ be artificial 1D images as shown in Figure 2, left and center left. Both images show three unique, corresponding objects. For the outer objects there is no change in position during transition from the template image T to the reference image R . The middle object changes its

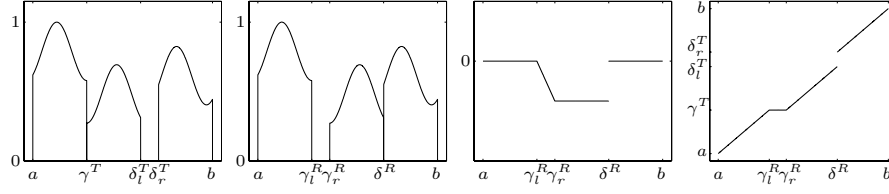


Fig. 2. Artificial 1D images T (left) and R (center left) which can be transformed into each other using the transformation (right) which employs the displacement u (center right).

position in such a way that a gap (represented by an interval of gray value 0) between it and the outer ones shows up or disappears.

Now, the task is to find a displacement u such that the transformation $x + u$ applied to the template image, i.e., $T(\cdot + u(\cdot))$, is similar to R .

One possible displacement u for mapping the images onto each other is shown in Figure 2, center right, together with the corresponding transformation aside. The discontinuity at δ^R is both responsible and necessary for closing the gap, given by $[\delta_l^T, \delta_r^T]$, to δ^R . Opening of a gap can be achieved by a ramp in the interval $[\gamma_l^R, \gamma_r^R]$ which maps all the positions located within this interval onto the position γ^T .

Nonetheless, this is not a unique transformation for the considered image pair. In general there are infinitely many transformations mapping the gray values onto each other. However, by construction the displayed transformation is monotone and therefore preferred over all other ones.

2.2 Standard Method

Though the synthetic example has been introduced as a 1D example, the variational approach is not restricted to one dimension. Rather, this approach can be formulated for any dimension.

Let $R, T : \Omega \rightarrow G$ denote the reference and the template image, respectively. Here, G denotes a set of grey values and $\Omega \subset \mathbb{R}^d$ the d -dimensional image domain. The registration aims at finding a displacement field $\mathbf{u} : \Omega \rightarrow \mathbb{R}^d$ such that $T_{\mathbf{u}} := T(\mathbf{id} + \mathbf{u})$ is similar to R , where $\mathbf{id}$ denotes the identity mapping. In mathematical terms, similarity is described by a functional $\mathcal{D}[\mathbf{u}; T, R]$. $\mathcal{D}$ can be chosen as any popular distance (or similarity) measure provided its Gâteaux derivative exists. However, this note is restricted to the common sum of squared differences,

$$\mathcal{D}[\mathbf{u}; T, R] := \int_{\Omega} [R(\mathbf{x}) - T_{\mathbf{u}}(\mathbf{x})]^2 d\mathbf{x}, \quad (1)$$

which assumes monomodal images.

A registration based on a similarity measure only, yields a deformed template image which perfectly matches the reference image as long as all gray

values are present in both images. However, it is well known that such a registration problem is ill-posed [9, 2]. Furthermore, the underlying deformation may be senseless in a physical context. Therefore, a smoothness constraint (or regularizer) $\mathcal{S}[\mathbf{u}]$ is added which can be chosen to model the application specific physical properties. In general it can be interpreted as a penalizer. In this note we investigate the so-called diffusive regularizer [9],

$$\mathcal{S}[\mathbf{u}] := \int_{\Omega} \sum_{i=1}^d \|\nabla u_i\|_2^2 d\mathbf{x}, \quad (2)$$

which penalizes oscillating deformations and leads to smooth deformation fields. It is named after the diffusion equation from physics, whose stationary case is equivalent to the Gâteaux derivative of (2). Other possible choices for the regularizing function include an elasticity- or a curvature-based approach, cf., e.g., [9].

Hence, a registration formulated as a variational model searches for a displacement field $\mathbf{u}$ minimizing a joint functional

$$\mathcal{J}[\mathbf{u}; T, R] := \mathcal{D}[\mathbf{u}; T, R] + \alpha \mathcal{S}[\mathbf{u}], \quad (3)$$

where α is a weighting factor controlling the influence of the smoothness term compared to the similarity measure (for further details, see [6]).

The computation of the Gâteaux derivative of (3) yields a necessary condition for $\mathbf{u}^*$ being a minimizer of (3). The outcome are nonlinear partial differential equations known as the nonlinear Poisson equations. Finally, a discretization by finite differences and a fixed-point type iteration scheme is applied to solve the set of partial differential equations. Here, the linear equation system is of size dN , N being the total number of voxels in Ω . The system matrix corresponds to the Poisson differential operator whereas the right hand side results from the similarity measure and may be seen as a force vector.

This scheme will be referenced as the standard method throughout this note.

2.3 Application to the Synthetic Example

In the one-dimensional case, the Gâteaux derivative of

$$\begin{aligned} \mathcal{J}[u; T, R] &:= \mathcal{D}[u; T, R] + \alpha \mathcal{S}[u] \\ &= \int_a^b [R(x) - T_u(x)]^2 dx + \alpha \int_a^b [u'(x)]^2 dx \end{aligned} \quad (4)$$

accompanied by the fundamental lemma of calculus of variations yields the following necessary condition for a minimizer, known as Euler-Lagrange equation,

$$\begin{aligned}
\alpha u''(x) &= f(x, u(x)) \quad \forall x \in [a, b], \\
u(a) &= u_a, \\
u(b) &= u_b,
\end{aligned} \tag{5}$$

with a right hand side abbreviated by

$$f(x, u(x)) := T'_u(x) [R(x) - T_u(x)], \tag{6}$$

which is nonlinear in the sought displacement u . Clearly, the boundary conditions can be chosen problem dependent. Here, conditions of Dirichlet type are employed.

Applying (5) to the synthetic example we obtain a result which is far away from the expected one, cf. Figure 3, center. The achieved displacement u (solid

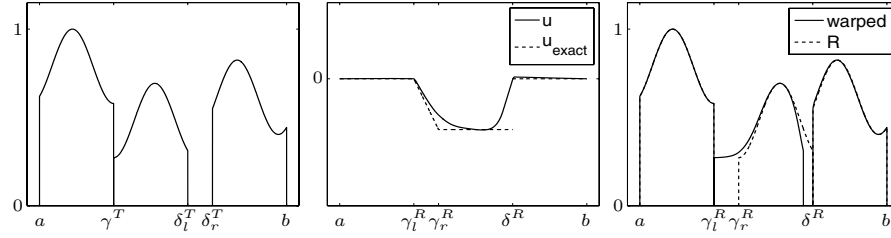


Fig. 3. Standard method applied to the synthetic example using a constant $\alpha = 100$: The displacement u (center) deforms the template image T (left) and yields a registration result which does not change the topology (right).

line) leads to a topology-preserving deformation of the template image (see solid line in Figure 3, right). Though, the gap interval $[\delta_l^T, \delta_r^T]$ has become smaller, it is neither closed nor a gap in $[\gamma_l^R, \gamma_r^R]$ has opened (for better comparison we added the expected function and the reference image – mind the dashed lines in Figure 3, center and right, respectively). This is due to the regularizer which penalizes oscillations of u . Therefore, a large difference in the gray values is less costly than a large gradient in the displacement. By decreasing α to a positive value $\ll 1$ a result similar to the expected one becomes more or less achievable. However, the regularizer is now weakened all over the image domain. The topology may change in regions other than wanted and image noise will start to influence the result. Therefore, instead of a global reduction of regularization, a locally diminished regularization is necessary. In particular, only regions with a gap expected to open or close need a local reduction of smoothness whereas all other regions shall be regularized according to the chosen physical model.

3 Variable Regularizer

To obtain a spatially dependent regularization, we replace the weighting factor α by a weighting function $\alpha : \Omega \rightarrow \mathbb{R}^+$, yielding

$$\begin{aligned} \mathcal{J}[\mathbf{u}; T, R] &:= \mathcal{D}[\mathbf{u}; T, R] + \mathcal{S}[\mathbf{u}; \alpha] \\ &= \int_{\Omega} [R(\mathbf{x}) - T_{\mathbf{u}}(\mathbf{x})]^2 d\mathbf{x} + \int_{\Omega} \alpha(\mathbf{x}) \sum_{i=1}^d \|\nabla u_i\|_2^2 d\mathbf{x} . \end{aligned} \quad (7)$$

Immediately, (7) requires a definition of the weighting function α . A priori it is not clear, if a specific choice of the weighting function can enforce a particular behavior of the solution such as a discontinuity in the displacement field itself or in its gradient. Next, one has to determine in which cases such a particular behavior is desired. Since it cannot simply be deduced from the gray values, further knowledge is required. Finally, we are interested in a minimizer of (7). Its existence and uniqueness are important issues and need investigation. However, their proofs are outside the scope of this note. Assuming the existence of a minimizer we will raise the issue of how the variable regularizer is related to the minimization task.

3.1 Choice of the Weighting Function

For choosing a weighting function α our purpose is twofold: First, we wish to model a closing gap as seen in the synthetic example (Figure 2). Two material objects with a gap inbetween are pushed together, the gap contracts to an infinitely small size indicating a closure. The second case is vice versa. Two objects being close to each other and divided by an infinitely small gap may be seen as connected but they can behave separately from each other. Applying a pulling force on the right object in the synthetic example yields a gap opening between the objects which are disconnected from then on. Therefore the question is, how to choose α in order to force the displacement field to exhibit a particular behavior. To this end, the analytical solution of a one-dimensional variational model will be inspected. Afterwards, we will construct sequences for α and u which converge to limit functions showing the desired discontinuity.

Let us note, that our aim is not to propose a one-dimensional registration method based on an analytical solution. Instead we are aiming at bringing forward insights from the analytical inspection of different weighting functions towards a generalized variational approach.

From variational calculus we know that (7) supplemented by boundary conditions of, for instance, Dirichlet-type, leads in the one-dimensional case to the Euler–Lagrange equation

$$\begin{aligned} [\alpha(x)u'(x)]' &= f(x, u(x)) \quad \forall x \in [a, b] , \\ u(a) &= u_a, \\ u(b) &= u_b, \end{aligned} \quad (8)$$

where f denotes the same term as in (6). To investigate the relation between α and u , an analytical solution of (8) is of great utility. However, (8) is nonlinear in u and in general the solution of a nonlinear (partial) differential equation cannot be given explicitly. Therefore, and since the relation between α and u is restricted to the left hand side, we set the right hand side to 0 which corresponds to the diffusion equation in the stationary case. For this reduced form an analytical solution is available. Let $(\alpha_n)_{n \in \mathbb{N}}$ denote a sequence of weighting functions and let $(u_n)_{n \in \mathbb{N}}$ be the sequence of solutions determined by (8) with zero right hand side.

We start with recalling the first case describing two material objects with a gap inbetween which is to be closed. Placing forces pushing the objects together corresponds to boundary conditions with $u_a < u_b$. Now, α_n is searched such that u_n converges to a step function as $n \rightarrow \infty$. Note that the variational model aims at minimizing the regularizer (in conjunction with the similarity measure). Thus a high value of u' at any position leads to a penalization of u . A sequence u_n , which exhibits a discontinuity at some position in its limit case, shows an increasing gradient and, therefore, requires a small weighting at the same position. Apart from that position, a differing weighting is not required. From Figure 4, left, we observe this property for a Dirac-shaped weighting function.

The second case needs a weighting function in order to let two formerly connected objects disconnect from each other. To this end, a pulling force on the right object is assumed from which we deduce boundary conditions with $u_a > u_b$. Choosing a cup-shaped weighting function we expect a displacement which is linear within the cup and exhibits a kink at each side. For $n \rightarrow \infty$ the desired ramp function is obtained (Figure 4, right).

For the actual implementation $\alpha_n(x) = 1/(1 + c_1 \exp(-nx^2))$ is chosen for the first case and $\alpha_n(x) = 1/(1 + c_2 \exp(-(c_3 x)^{c_4 n}))$ is chosen for the second case, both with appropriate constants (see Figure 4, top).

We are thus able to choose a specific weighting function such that u according to (8) displays the wanted behavior. It remains to be determined, in which cases u shall be forced to display one of these behaviors. Further knowledge is required and given by a segmentation.

3.2 Segmentation

A meaningful segmentation of the images into anatomical regions (and background) needs to be computed. That is, we are looking for disjoint regions Ω_l , such that $\Omega = \cup_{l=0}^m \Omega_l$ and each region Ω_l^R in R corresponds intrinsically to a region $\Omega_l^{T_u}$ in image T_u . For convenience, let $\Omega_0^R, \Omega_0^{T_u}$ denote the background of image R and image T_u , respectively. Such a segmentation may not be easy to obtain. However, its computation is outside the scope of this note.

Assuming a segmentation, we are then in a position to let discontinuities be introduced, if these appear to be physically meaningful. More precisely,

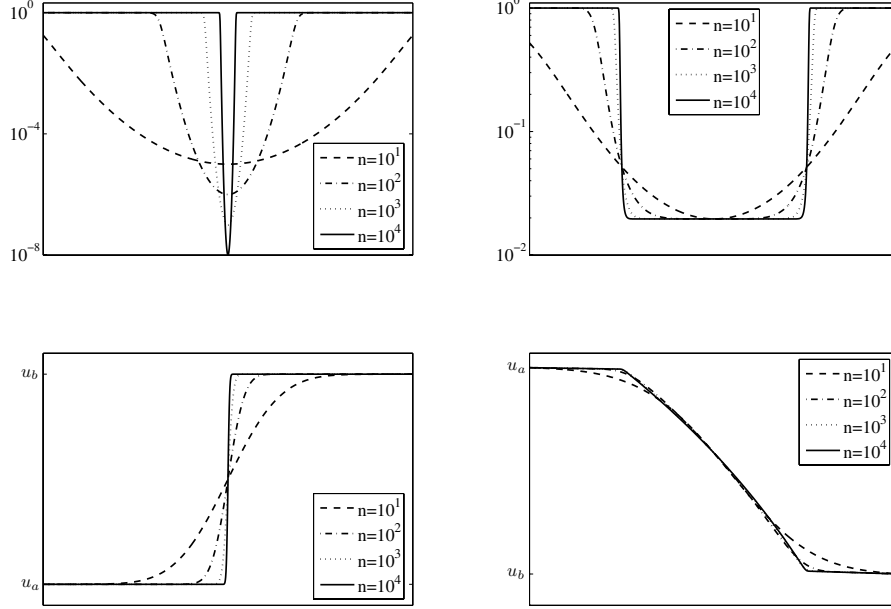


Fig. 4. Relation between weighting functions α (top row) and displacement functions u (bottom row) for the step case (left) and the ramp case (right).

- a Dirac-shaped weighting function is set along a non-overlapping interface between two regions Ω_{l_1} and Ω_{l_2} , where non-overlapping regions can be detected by $\Omega_{l_1}^R \cap \Omega_{l_2}^{T_u} = \emptyset$ and $\Omega_{l_2}^R \cap \Omega_{l_1}^{T_u} = \emptyset$. If, in addition, the forces induced by $\mathcal{D}$ in $\Omega_{l_1}^{T_u}$ and $\Omega_{l_2}^{T_u}$ along the interface do not point in the same direction, a discontinuity in the displacement field will arise,
- a cup-shaped weighting function is set in any subregion turning from foreground to background during transition from T to R , i.e., in $\cup_{l=1}^m \Omega_l^{T_u} \cap \Omega_0^R$.

To evaluate the deformed template image $T_{\mathbf{u}}(\mathbf{x})$, an interpolation scheme has to be exploited. The interpolation, again, has to be done with respect to the segmentation. The grey value $T_{\mathbf{u}}(\mathbf{x})$ is calculated by a linear interpolation of $\mathbf{x} + \mathbf{u}(\mathbf{x})$ with respect to T provided that $\mathbf{x}$ and $\mathbf{x} + \mathbf{u}(\mathbf{x})$ belong to corresponding segmentation regions. Otherwise $T_{\mathbf{u}}(\mathbf{x})$ is set to some prescribed background value. This situation indicates an infinitely small gap which is not visible in the discrete image.

3.3 Consequences for the Minimizing Functional

Obviously, using a constant α , u according to Figure 2 does not minimize (4). Any slightly changed u_ε with a smaller gradient than in u yields a smaller value of $\mathcal{J}$.

To make a statement for a weighting function α requires the determination of the global minimum in a function space $U \equiv C^4$ consisting of all four times continuously differentiable functions. This is, however, not possible in general. In exchange we consider a parametric solution space $U_p \subset U$ consisting of all smoothed functions of the form as shown in Figure 2, center right. Every $u \in U_p$ is determined by a parameter vector $p \in \mathbb{R}^3$ describing the positions of the ramp and the step as well as its amplitude. An exhaustive search through the solution space U_p with respect to the synthetic example returns as the minimum a displacement $u(p)$ which describes exactly the smoothed version of the wanted displacement from Figure 2. Though this is clearly not ensured in general, in this context it is plausible that $u(p)$ minimizes $\mathcal{J}$ with respect to U as well.

Clearly, in the proposed method the exhaustive search in the (highly restricted) solution space U_p is not employed. Instead, it is replaced by a numerical solution of the underlying Euler–Lagrange equation with respect to the (general) solution space U .

3.4 Discretization

Until now we assumed both the weighting function and the displacement field to be as often continuously differentiable as needed. Large gradients as they occur in sequences converging to a step or ramp in the limit case are, however, still differentiable but may lead to higher discretization errors in the numerical solution part. Furthermore, the limit case $n \rightarrow \infty$ cannot be described by the approach proposed so far due to the lack of differentiability.

Consequently we distinguish two alternatives in the following. The first assumes α and u to be differentiable everywhere in the image domain $[a, b]$. The second alternative loosens the previous assumption and allows α and u to be non-differentiable at a single position γ .

- First alternative: $U = C^4$
Let the solution space U consists of all four times continuously differentiable functions defined on the interval $[a, b]$, i.e., $U := C^4([a, b], \mathbb{R})$. By variational calculus we arrive at the Euler–Lagrange equation already given in (8). A discretization using central finite differences usually yields a method with a local truncation error of second order. Here, an approximation of first order can be achieved only. This is due to the unbounded derivative of α .
- Second alternative: $U = PC^4$
Let γ be some position inside the image domain, i.e., $a < \gamma < b$. Then, by $U := PC^4(\{[a, \gamma[,]\gamma, b], \mathbb{R}\})$ we define the solution space as the set of functions which are piecewise, i.e., on both intervals, included in the space C^4 . Since α and u are not differentiable at γ , the calculation of the Gâteaux derivative needs some care. The similarity measure does not include derivatives of α or u , therefore we focus our attention on the regularizing term.

Taking the integral of $\alpha(x)[u'(x)]^2$ over both intervals $[a, \gamma[$ and $] \gamma, b]$ separately, we omit the null set where u' is not defined. This does not change the value of the integral. Then, the necessary condition for a stationary point of $\mathcal{S}$ requires

$$- \int_{x \in [a, b] \setminus \gamma} [\alpha(x)u'(x)]' v(x) dx + \{(\alpha u'v)(\gamma-) - (\alpha u'v)(\gamma+)\} + \{(\alpha u'v)(b) - (\alpha u'v)(a)\} = 0 \quad (9)$$

to hold for all test functions $v \in U$. By employing natural boundary conditions on a and b , the second curly bracketed term vanishes. In order to apply the lemma of variation, the first curly bracket has to be zero as well. We call the resulting additional condition

$$(\alpha u'v)(\gamma-) - (\alpha u'v)(\gamma+) = 0, \quad (10)$$

motivated by the analogy to natural boundary conditions, a natural interface condition. In conjunction with the Gâteaux derivative of the similarity measure, by the lemma of variation and due to α being always positive, we finally arrive at the Euler–Lagrange equation

$$\begin{aligned} [\alpha(x)u'(x)]' &= f(x, u) \quad \forall x \in [a, b] \setminus \gamma, \\ u'(\gamma-) &= u'(\gamma+) = 0, \\ u(a) &= u(b) = 0. \end{aligned} \quad (11)$$

The major difference to the first alternative is given by the additional interface condition at γ . Again, a discretization using central finite differences is employed. This time we obtain a method with a local truncation error of second order.

A comparison of the resulting discretized equations from both alternatives reveals that they almost coincide. In fact, they are consistent with the differential equations from both alternatives but represent the second alternative with a local truncation error of higher order. As described in Subsect. 2.2 the discretized equations yield a linear equation system. Now, the system matrix includes the additional information given by the segmentation and by the weighting function as well.

4 Numerical Results

The proposed method has been applied to the synthetic example as well as to real-life images.

4.1 Synthetic Example

As it is apparent from Figure 3, a standard registration approach fails in the case of the synthetic example. In particular the gap regions have been penalized in such a way that a large difference in the gray values are preferred over a large gradient in the displacement. As a consequence neither the gap on the right side of the moving object closed totally nor did the gap on its left side open.

The variable diffusive registration approach proposed in Sect. 3 can be expected to cope with the changing topology. We start with the image pair from Figure 2 supplemented by a segmentation of both images into three subdomains each: Each object belongs to a single region Ω_l . The weighting function α is chosen according to Subsect. 3.2 (see Figure 5, left).

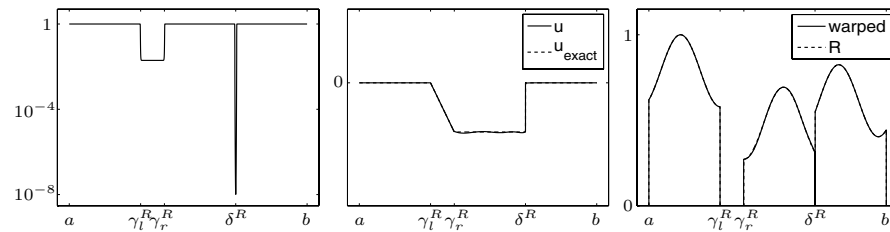


Fig. 5. Synthetic example after registration with the proposed method. An appropriate weighting function (left) leads to a displacement function (center) and a warped image (right), which fit nicely the exact displacement and the reference image, respectively.

Comparing the new result (Figure 5) to the previous one (Figure 3) the outcome is twofold. First, a change in topology has been achieved: A gap can open as well as contract totally. Second, the warped image as the registration result fits nicely to the reference image. The differences between both the exact/recovered displacement and the warped/reference image are negligible.

4.2 MR Head Images

As a real-life example the corresponding head slices (170×195 , voxel size is $1 \times 1 \text{ mm}^2$) from Figure 1 have been registered. Again, registration is performed using the standard scheme with no spatially varying regularization and with the proposed method based on a manual segmentation of the head (cf. Figure 6). Both images consist of three subdomains: brain, skull and skin area. Additionally, the intra-operatively taken image includes a background region between the subdomains covering brain and skull, shown in black in Figure 6, center right.

Comparing the deformed images an improved registration result is nicely visible. In particular in the region affected by the brain shift a displacement

in different directions occurs (cf. the reference and the template image in Figure 6). While the standard scheme fails for this task, whatever a value for α is chosen, the proposed method yields an inward displacement of the brain and, independently from that, a displacement of the skull/skin area.

The similarity measure for the standard approach reduces to 50% compared to the original value whereas for the proposed method, a reduction to 19% is achieved. To some extent, this further reduction is caused by the skin which is partly removed in the reference image. Whereas the displacement field in this area is smoothed in the standard scheme, the skin is allowed to shrink independently from the other structures in our new approach. For a more detailed analysis of the deformation fields we manually placed markers in the brain/skull area. For twelve markers in the brain the standard and the proposed method yield a median difference of 1.87 mm and 0.74 mm, respectively, compared to the markers in the reference image. For three markers placed in the skull the difference is even higher: 4.65 mm for the standard method compared to 0.83 mm for our proposed method.

5 Conclusion and Outlook

We have introduced a diffusion matching based registration scheme with a spatially varying regularization. The properties of a variable regularizer show its ability for discontinuities in the displacement field in order to cope for topological changes between template and reference image.

The feasibility of the proposed algorithm has been shown for a synthetic example as well as for a nontrivial real-life example. The first results are very convincing and superior compared to standard schemes, when pathological changes occur in the images.

An extension to a regularizer based on the linear elastic potential is in progress. From the combination of an elastic approach and a variable regularizer we expect a deformation field which is better suited to applications with deformable elastic tissues.

Also it would be desirable, if only one of the images needs to be segmented. In particular, for time-critical tasks, like brain shift, it is of great interest if a segmentation of the pre-operatively generated image is sufficient. If so, a time consuming segmentation of the intra-operatively generated image would be redundant. This work is in progress.

Acknowledgement. We are grateful to Dr. Hastreiter (Department of Neurosurgery, Friedrich Alexander University of Erlangen–Nuremberg) for providing the MR images.

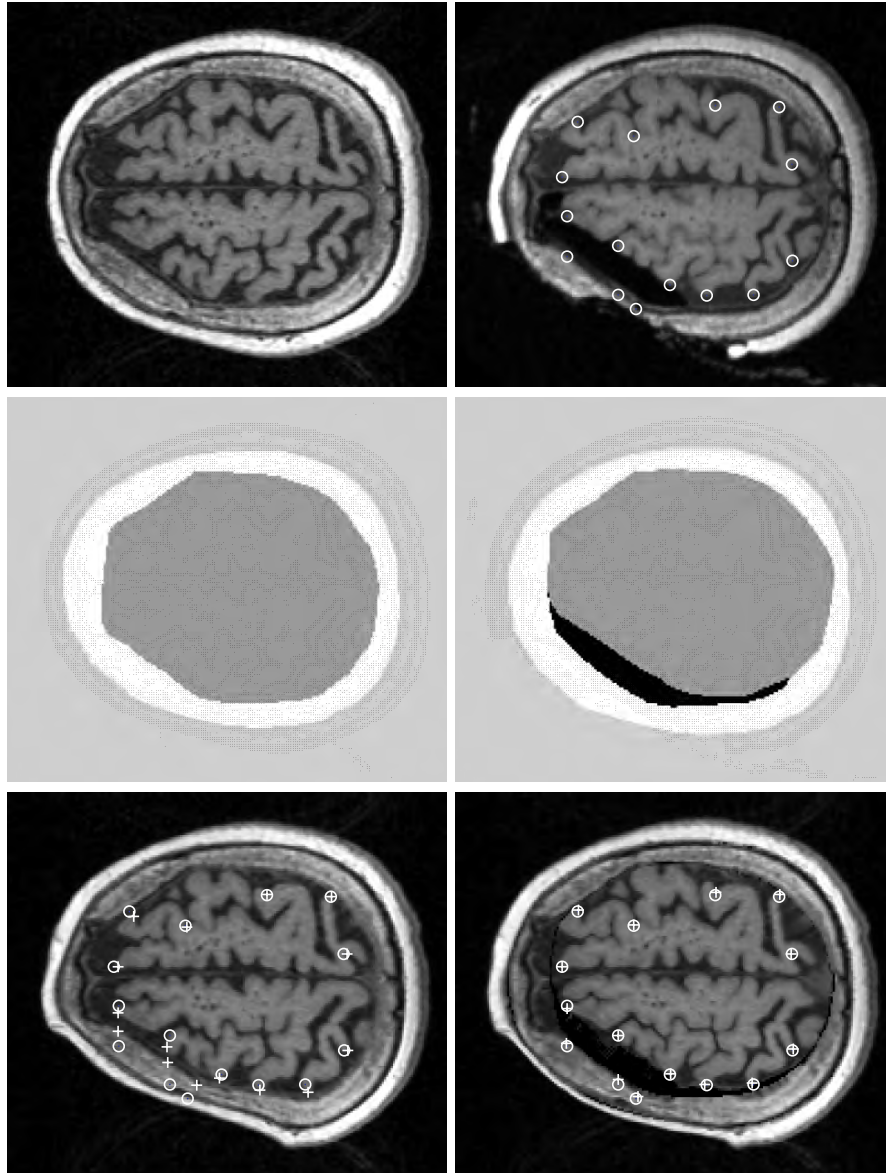


Fig. 6. MR head images (upper row) with their segmentations (center row, for convenience underlayed with the corresponding edge images) and after registration (lower row). Registration is performed using a standard approach (lower left image) as well as the proposed method (lower right image).

References

1. R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Comput. Vis. Graph. Image Process.*, 46(1):1–21, 1989.
2. U. Clarenz, M. Droske, S. Henn, M. Rumpf, and K. Witsch. Computational methods for nonlinear image registration. 2005. to appear.
3. C. Davatzikos. Nonlinear registration of brain images using deformable models. In *Proc. of the IEEE Workshop on Math. Methods in Biomedical Image Analysis*, pages 94–103, 1996.
4. P.J. Edwards, D.L.G. Hill, J.A. Little, and D.J. Hawkes. A three-component deformation model for image-guided surgery. *Med. Image Anal.*, 2(4):355–367, 1998.
5. M. Ferrant, A. Nabavi, F.A. Jolesz, R. Kikinis, and S.K. Warfield. Registration of 3d intraoperative mr images of the brain using a finite element biomechanical model. *IEEE Trans. Med. Imag.*, 20(12):1384–1397, 2001.
6. B. Fischer and J. Modersitzki. Fast inversion of matrices arising in image processing. *Numer. Algorithm.*, 22:1–11, 1999.
7. A. Hagemann. *A Biomechanical Model of the Human Head with Variable Material Properties for Intraoperative Image Correction*. Logos, Berlin, 2001.
8. S. Kabus, T. Netsch, B. Fischer, and J. Modersitzki. B-spline registration of 3D images with Levenberg–Marquardt optimization. In *Proc. of SPIE 2004*, volume 5370, pages 304–313, 2004.
9. J. Modersitzki. *Numerical methods for image registration*. Oxford University Press, 2004.
10. J. Rexilius, H. Handels, A. Nabavi, R. Kikinis, and S.K. Warfield. Automatic nonrigid registration for tracking brain shift during neurosurgery. In *Proc. of BVM 2002*, pages 135–138, 2002.
11. G.K. Rohde, A. Aldroubi, and B.M. Dawant. The adaptive bases algorithm for intensity-based nonrigid image registration. *IEEE Trans. Med. Imag.*, 22(11):1470–1479, 2003.
12. G. Soza, P. Hastreiter, F. Vega, C. Rezk-Salama, M. Bauer, C. Nimsy, and G. Greiner. Non-linear intraoperative correction of brain shift with 1.5 T data. In *Proc. of BVM 2003*, pages 21–25, 2003.

Part VI

Inverse Problems

Shape Reconstruction from Two-Phase Incompressible Flow Data using Level Sets

Rossmaty Villegas, Oliver Dorn, Miguel Moscoso, and Manuel Kindelan

Grupo de Modelización y Simulación Numérica, Universidad Carlos III de Madrid,
Avenida de la Universidad 30, Leganes 28911, Spain. E-mail:
`rvillega@math.uc3m.es`

Summary

We present a novel level set technique for shape reconstruction in history matching for non-conventional reservoirs. These reservoirs consist of several regions with different materials, e.g., shale or sand. The goal is to use the production data in order to estimate the unknown shapes and structure of these regions in the reservoir. Mathematically, we formulate this situation as an inverse problem for a non-compressible two-phase flow equation, describing the propagation of oil and water in the reservoir. The shapes or regions (which in our case consist of either sand or shale) are represented by a level set function which needs to be determined from the production data. We present numerical results in 2D which demonstrate that our method is able to provide reliable estimates of this structure from relatively few production data even though the topology of the unknown regions is a-priori unknown.

1 Introduction

In reservoir engineering for secondary oil recovery, typically water is injected in several injection wells with the goal to enhance oil production in other production wells. During this so-called ‘water flooding’ process, the injected water flows from the injection wells towards the production wells. The corresponding oil-water flow can then be modelled as a two-phase incompressible flow problem in a porous medium. (We will neglect here the possible presence of gas, which would give rise to a more realistic but also more complicated compressible three-phase flow model).

In order to optimize the production process, the reservoir engineer needs to understand the geological structure inside the earth, such that this knowledge can be incorporated into the numerical reservoir simulators. Typically, only very few data are available which can be used for establishing this essential information, such that there is high uncertainty in the characterization of the

reservoir. One useful source of information is for example obtained by so-called ‘well-logs’, which consist of measurements taken inside each of the wells. These provide the reservoir engineer with mainly local information of the earth in the area surrounding these wells. In addition, some general geological information can be deduced from these well-log data. Other additional sources for information are seismic and electromagnetic measurements obtained in specifically designed physical experiments. However, the corresponding procedure of measuring these data is expensive and, in addition, these additional data might not contain sufficient information in order to completely characterize the reservoir. Therefore, throughout the production it is attempted to use directly the production data, for example the pressure and/or the water flow rate at the production wells, for inferring additional information about the physical parameters inside the active reservoir. This procedure of determining physical parameters of the reservoir which ‘match’ the production data when put into the reservoir simulator is usually called ‘history matching’. The corresponding inverse problem is severely ill-posed and difficult to solve.

Due to its great importance in oil production, the history matching problem has a long history, and many solution techniques have been developed to date. One possibility for solving the reservoir characterization problem is to put it into the framework of optimal control or constraint optimization problems, see, e.g., [1, 4, 6, 9, 10, 12, 13, 22, 23]. Given the measured data, physical parameter distributions inside the reservoir are sought in these approaches which minimize a suitable chosen cost functional. One severe difficulty in these approaches is given by the mentioned sparsity of the available data and the severe ill-posedness. In order to cope with these difficulties, typically very strong regularization techniques need to be applied, which are designed to stabilize the inversion process and to yield a well-defined solution.

In many practical situations, the reservoir engineer has quite a good understanding about the general geological composition of the reservoir (e.g., by interpreting well-log data), which can be incorporated into the mathematical model of the reservoir [3, 11]. For example, in some situations it is known that the reservoir consists mainly of two materials, shale and sand, with relatively well-known physical properties. Therefore, quite recently the possibility has been discussed to incorporate this prior information of a binary nature of the physical parameters into the reservoir model, in order to improve the stability and quality of the parameter estimation from production data [18, 19]. In this approach, the inverse problem can be reduced to a shape reconstruction problem which only tries to reconstruct the boundaries of the different regions (e.g. filled with shale or sand) from the data.

Classical pixel- or voxel-based inversion techniques usually do not incorporate this type of prior information into the inversion process. Instead, they try to reconstruct for each pixel or voxel an individual physical parameter value, such that each pixel/voxel can in principle consist of a different physical material. Such an approach has been presented for example in [10]. Standard regularization tools (like Tikhonov regularization) try to avoid a small-scale

fragmentation of the reservoir reconstruction by strongly smoothing the parameter distributions. This, however, has the side-effect that physical discontinuities in the reservoir are as well smeared out significantly over a large area. The separation into different zones of known material parameters must then be done by applying postprocessing tools. For example, standard image segmentation techniques can be applied on the smooth images in order to separate the reconstructed pixel- or voxel-based representations into zones of known physical parameters. However, during the segmentation of the images, the effort of fitting the data during the solution of the inverse problem is partly undone, since image segmentation techniques change the images without taking into account production data. It would be desirable to have tools available which incorporate the additional information about the binary nature of the physical parameters directly into the inversion process, without the need for a postprocessing step as described above.

In this paper we will present a newly developed algorithm which tries to provide such a tool. We treat the history matching problem as a shape reconstruction problem for two-phase incompressible flow of oil and water in a porous medium. We use a level set technique [2, 5, 8, 14, 15, 16, 17, 20, 21] for modeling the different regions of the earth which are filled with two different physical materials, in our case shale and sand. The topology of these regions is a-priori unknown. Using the available production data, and some prior information from well-logs affecting physical parameters directly at the wells, we try to recover a binary map of permeability values in the reservoir.

In this model, the permeability in the earth is assumed to have essentially two different values, one value for sand material and one value for shale. These might be average values determined for the given reservoir. We construct an artificial shape evolution modeled by the level set function which tries to reduce the mismatch between calculated data (corresponding to the actual topology of the reservoir at the given stage of the evolution) and the physically measured data. A very useful property of the level set technique during this evolution is the fact that we do not need to know a-priori the topology of the sand/shale distribution in the reservoir. The level set representation will automatically change the topology during the artificial shape evolution, if necessary, in order to fit the production data.

We mention that an interesting history matching approach based on level sets has been presented very recently in [2, 14]. In that work, a multi-levelset representation of the shapes combined with a multi-scale technique for regularizing the reconstruction is used for recovering permeability structures in the reservoir. Our scheme differs in several aspects from that approach. Firstly, we apply a so-called ‘adjoint scheme’ calculating shape sensitivities during the reconstruction, which is known to be very efficient for large-scale inversion problems. Secondly, as regularization tool we use a specifically adapted filtering operator which needs to be applied to the updates in each step of the inversion process. This is an alternative to applying multi-scale techniques as done in [2, 14]. This combination of the adjoint scheme and the novel regularization

scheme allows us to characterize reasonably sized 2D reservoirs, where several wells are present, on a regular PC in a relatively short time. All forward modeling codes which are used in our numerical simulations (A streamline technique as well as an IMPES technique for modelling incompressible two-phase flow in the porous medium) have been developed and implemented by our research group ‘Modelization and Numerical Simulation’ at University Carlos III de Madrid. However, the inversion algorithm presented here is not restricted to the use of these codes, and can as well be applied using standard commercial 2D or 3D reservoir simulators for performing the forward modelling task, as long as they can be combined efficiently with the available adjoint solvers.

We will present in this paper numerical experiments for realistic 2D situations which show that our shape-based inversion technique is able to recover shapes with quite complicated topology in a stable way from only few production data. These data have been simulated with an independent reservoir simulator in order to avoid the so-called ‘inverse crime’.

The paper is organized as follows. In Section 2 our simplified model is introduced for two-phase flow in porous media. In Section 3 the mathematical forward problem is derived, and in Section 4 the inverse problem is formulated as a shape reconstruction problem. In Section 5 we derive formally the shape evolution algorithm using level sets. In Section 6 we derive efficient practical ways of calculating sensitivities or Frechet derivatives for our model using an adjoint technique. In Section 7 we summarize the algorithm for shape reconstruction used in our work, and in Section 8 we present two numerical test cases in 2D which demonstrate the practical performance of the algorithm in realistic situations. Finally, in Section 9, we present some conclusions and give indications for future research.

2 The Reservoir Model

Our simplified model for two-phase flow in porous media for reservoir engineering is given as

$$\phi \frac{\partial S_w}{\partial t} - \nabla \cdot [T_w (\nabla p_w + \rho_w g \mathbf{k})] = Q_w \quad \text{in } \Omega \times [0, t_f] \quad (1)$$

$$\phi \frac{\partial S_o}{\partial t} - \nabla \cdot [T_o (\nabla p_o + \rho_o g \mathbf{k})] = Q_o \quad \text{in } \Omega \times [0, t_f]. \quad (2)$$

These two conservation laws for water (subscript w) and oil (subscript o), considered as incompressible fluids in a porous medium, are typically augmented by the two additional equations

$$P_{cwo} = p_o - p_w, \quad (3)$$

$$S_w + S_o = 1. \quad (4)$$

This yields four equations (1)–(4) in the four unknowns p_w , p_o , S_w and S_o . Hereafter, the subindex ‘ w ’ stands for ‘water’, and the subindex ‘ o ’ stands for ‘oil’. Equation (3) links the water and oil pressures (p_w and p_o , resp.) in the medium by the capillary pressure P_{cwo} . Equation (4) links the saturations S_w of water and S_o of oil and indicates that the porous medium is fully saturated. Gravity effects are taken into account by the terms $\rho_w g \mathbf{k}$ and $\rho_o g \mathbf{k}$. These two terms, together with the capillary pressure P_{cwo} , are incorporated in our forward modeling code, but are assumed to be small and will be neglected when deriving the algorithm for solving the inverse problem. $\Omega \subset \mathbf{R}^n$ ($n = 2, 3$) is the modeling domain with boundary $\partial\Omega$, and $[0, t_f]$ is the time interval for which production data is available. We denote by $\phi(\mathbf{x})$ the porosity, and by T_o , T_w and T the transmissibilities, which are known functions of the permeability K and the water saturation S_w :

$$T_w = K(\mathbf{x}) \frac{K_{rw}(S_w)}{\mu_w}; \quad T_o = K(\mathbf{x}) \frac{K_{ro}(S_w)}{\mu_o}; \quad T = T_w + T_o. \quad (5)$$

Here, the relative permeabilities $K_{rw}(S_w)$ and $K_{ro}(S_w)$ are typically available as tabulated functions, and μ_w and μ_o denote the viscosities of each phase. Q_o , Q_w and $Q = Q_o + Q_w$ define the oil flow, the water flow and the total flow, respectively, which are measured at the well positions. Equations (1)–(4) are solved with appropriate initial conditions, and a no-flux boundary condition on $\partial\Omega$.

When neglecting the gravity terms $\rho_w g \mathbf{k}$ and $\rho_o g \mathbf{k}$, as well as capillary pressure (such that $p_w = p_o = p$), equations (1)–(4) simplify to the two equations

$$-\nabla \cdot [T \nabla p] = Q \quad \text{in } \Omega \times [0, t_f] \quad (6)$$

$$\phi \frac{\partial S_w}{\partial t} - \nabla \cdot [T_w \nabla p] = Q_w \quad \text{in } \Omega \times [0, t_f] \quad (7)$$

for the two unknowns p and S_w , which we supply with the following initial and boundary conditions

$$S_w(\mathbf{x}, 0) = S_w^0(\mathbf{x}) \quad \text{in } \Omega, \quad (8)$$

$$p(\mathbf{x}, 0) = p^0(\mathbf{x}) \quad \text{in } \Omega, \quad (9)$$

$$\nabla p \cdot \boldsymbol{\nu} = 0 \quad \text{on } \partial\Omega. \quad (10)$$

Here, $\boldsymbol{\nu}$ is the outward unit normal to $\partial\Omega$. The boundary condition (10) implies no flux across the boundary. Equations (6)–(10) will be our basic model for deriving the shape inversion algorithm. $Q(\mathbf{x}, t)$ and $Q_w(\mathbf{x}, t)$ define the total flow and the water flow at the wells, respectively. They are given by

$$Q = cT \sum_{j=1}^{N_i} (p_{wb_j}^{(i)} - p) \delta(\mathbf{x} - \mathbf{x}_j^{(i)}) + cT \sum_{j=1}^{N_p} (p_{wb_j}^{(p)} - p) \delta(\mathbf{x} - \mathbf{x}_j^{(p)}) \quad (11)$$

$$Q_w = cT \sum_{j=1}^{N_i} (p_{wb_j}^{(i)} - p) \delta(\mathbf{x} - \mathbf{x}_j^{(i)}) + cT_w \sum_{j=1}^{N_p} (p_{wb_j}^{(p)} - p) \delta(\mathbf{x} - \mathbf{x}_j^{(p)}) \quad (12)$$

where $\mathbf{x}_j^{(i)}$, $j = 1, \dots, N_i$, denote the locations of the N_i injector wells, $\mathbf{x}_j^{(p)}$, $j = 1, \dots, N_p$, denote the locations of the N_p production wells, and $p_{wb_j}^{(i)}$, $p_{wb_j}^{(p)}$ are the imposed well bore pressures at the N_i injector wells and at the N_p production wells, respectively. Here, c is a constant that depends on the well model [7]. Since $p_{wb_j}^{(i)}$ ($p_{wb_j}^{(p)}$) are larger (smaller) than the reservoir pressure at the injector (production) wells, Q and Q_w are positive (negative) at the injector (production) wells.

3 The Forward Problem

We can now introduce the forward operator of our problem. We write equations (6)-(12) in operator form as

$$\Lambda(K)u = \mathbf{q} \quad (13)$$

with $u = (p, S_w)$ and where the right hand side $\mathbf{q}$ is defined by the right hand sides of (6), (7). Notice that the derivation of our algorithm is not restricted to the assumption of wells modeled by point sources, but that more complex descriptions of the wells (e.g., as interior boundary conditions) can easily be incorporated in the algorithm. We can define the forward operator Λ mapping the parameter K to the corresponding data $g = Mu$ by

$$\mathcal{A}(K) = Mu = M\Lambda(K)^{-1}\mathbf{q} \quad (14)$$

where M is the measurement operator given by

$$Mu = \{Q_{w,j}^{(p)}\}_{j=1 \dots, N_p}, \quad (15)$$

being the water flow obtained at the production wells. Practically, calculating $\Lambda(K)^{-1}\mathbf{q}$ means to run our reservoir simulator on the applied input pressure data with the permeability given as K . We will denote the physically measured ‘true data’ by

$$\tilde{g} = M\tilde{u}, \quad (16)$$

where $\tilde{u}$ denotes the (unknown) physical state given the correct parameter distribution $\tilde{K}$. Finally, we introduce the ‘residual operator’ $\mathcal{R}$ by defining

$$\mathcal{R}(K) = \mathcal{A}(K) - \tilde{g}. \quad (17)$$

4 The Shape Reconstruction Problem

In the shape inverse problem we assume now that the parameter K has the following specific form

$$K(\mathbf{x}) = \begin{cases} K_i & \text{in } D \\ K_e & \text{in } \Omega \setminus D. \end{cases} \quad (18)$$

That means, the domain of interest is divided in several disjoint zones inside which the parameters can only assume one of the two prescribed constant values (K_i or K_e). (Generalizations to smoothly varying profiles $K_i(\mathbf{x})$ and $K_e(\mathbf{x})$ are possible but will not be considered here.)

In the situation described here we want to identify the distribution or geological shapes of two lithofacies in the reservoir. We assign a mean permeability K_e for the first lithofacie and a permeability K_i for the second one. The unknown here is the geometric shape of each lithofacie. The procedure starts with a first approximation to the permeability distribution which in practice can be obtained from the available prior information given by well logs, well tests, core analysis or seismic interpretation. This distribution is mapped to a grid of $N_x \times N_y$ elements. From this initial guess, the algorithm computes a series of shapes which improve successively the match to the production data $\tilde{g}$.

To define the region D with the level set technique (for general information on the level set technique we refer to [16, 17, 21]), we introduce a sufficiently smooth level set function ψ such that

$$K(\mathbf{x}) = \begin{cases} K_i, & \text{if } \psi(\mathbf{x}) \leq 0 \\ K_e, & \text{if } \psi(\mathbf{x}) > 0. \end{cases} \quad (19)$$

Clearly, for each level set function ψ there is a unique region D associated with it. However, a given region D may be associated with different level set functions. The boundary of D (denoted as $\Gamma = \partial D$) is defined by the zero level set of the level set function ψ . To solve the shape reconstruction problem, we will adopt a time evolution approach [20]. As a consequence, Γ and ψ will be functions of an artificial evolution time t ,

$$\Gamma(t) = \{x : \psi(x, t) = 0\}. \quad (20)$$

In this approach, the inverse problem can be stated as follows: find a function ψ in (19) for which the least squares cost functional

$$\mathcal{J}(\psi) = \frac{1}{2} \|\mathcal{R}(\psi)\|^2 \quad (21)$$

is minimized. Notice that, slightly abusing notation, in (21) we used the symbol $\mathcal{R}$ also for the newly defined residual operator

$$\mathcal{R}(\psi) = \mathcal{R}(K(\psi)) \quad (22)$$

which now depends on the level set function ψ . It describes the mismatch between the actual water production data, $\tilde{g}$, and the water production data g obtained by solving the direct problem (6)-(12) for (19).

5 Formal Derivation of the Shape Evolution Algorithm

We want to derive a shape evolution which solves our above stated inverse problem. For this purpose, we consider the general evolution law

$$\frac{d\psi}{dt} = f(\mathbf{x}, t, \psi, \mathcal{R}, g, \tilde{g}, \dots) \quad (23)$$

for the level set function ψ describing the shape D during the artificial evolution. The goal is to find a forcing term $f(\mathbf{x}, t, \psi, \mathcal{R}, g, \tilde{g}, \dots)$, which might depend on a variety of parameters as indicated, such that the evolution converges to the desired solution of the inverse problem.

We introduce the one-dimensional Heaviside function $H(\psi)$ which is defined as

$$H(\psi) = \begin{cases} 1 & , \quad \psi > 0 \\ 0 & , \quad \psi \leq 0. \end{cases} \quad (24)$$

Then, we can write (19) as

$$K(\psi) = K_e H(\psi) + K_i (1 - H(\psi)). \quad (25)$$

Formally, differentiating (25) with respect to ψ yields

$$\frac{dK}{d\psi} = (K_e - K_i) \delta(\psi) \quad (26)$$

where $\delta(\psi) = H'(\psi)$ is the one-dimensional Dirac delta distribution. Furthermore, differentiating the least squares cost functional $\mathcal{J}(K(\psi(t)))$ with respect to the artificial time variable t and applying the chain rule yields

$$\frac{d\mathcal{J}}{dt} = \frac{d\mathcal{J}}{dK} \frac{dK}{d\psi} \frac{d\psi}{dt} = \left\langle \mathcal{R}'(K)^* \mathcal{R}(K), \frac{dK}{d\psi} \frac{d\psi}{dt} \right\rangle_P \quad (27)$$

where $\langle \cdot, \cdot \rangle_P$ denotes the inner product in our parameter space P . Plugging (23) and (26) into (27) yields finally

$$\frac{d\mathcal{J}}{dt} = \left\langle \mathcal{R}'(K)^* \mathcal{R}(K), (K_e(\mathbf{x}) - K_i(\mathbf{x})) \delta(\psi) f(\mathbf{x}, t, \psi, \mathcal{R}, g, \tilde{g}, \dots) \right\rangle_P. \quad (28)$$

Let us assume that the shape D is represented by a *continuously differentiable level set function* ψ such that $|\nabla\psi| \neq 0$ at the boundary of the shape. Then, we can use the relation

$$\delta(\psi) = \frac{\delta_{\partial D}(\mathbf{x})}{|\nabla\psi(\mathbf{x})|} \quad (29)$$

where $\delta_{\partial D}$ is the n -dimensional Dirac delta distribution concentrated on ∂D . Plugging this into (28) yields

$$\frac{d\mathcal{J}}{dt} = \left\langle \mathcal{R}'(K)^* \mathcal{R}(K), (K_e(\mathbf{x}) - K_i(\mathbf{x})) \frac{\delta_{\partial D}(\mathbf{x})}{|\nabla \psi(\mathbf{x})|} f(\mathbf{x}, t, \psi, R, g, \dots) \right\rangle_P. \quad (30)$$

Let us define now the steepest descent direction f_{SD} by

$$f_{\text{SD}}(\mathbf{x}, t, \psi, \mathcal{R}, g, \tilde{g}, \dots) = -F_{\text{SD}} |\nabla \psi| \quad (31)$$

with

$$F_{\text{SD}}(\mathbf{x}) = (K_e - K_i) \mathcal{R}'(K)^* \mathcal{R}(K) \quad \text{for } \mathbf{x} \in \partial D. \quad (32)$$

Then, (23) gets the form of a Hamilton-Jacobi-type equation

$$\frac{\partial \psi}{\partial t} + F_{\text{SD}} |\nabla \psi| = 0. \quad (33)$$

Notice that (31) is so far only defined on the boundary Γ of the shape D , such that we will need to determine a suitable ‘extension velocity’ in order to solve (23). Regardless which extension velocity we choose, (30) becomes for this case

$$\begin{aligned} \frac{d\mathcal{J}}{dt} = & - \int_{\partial D} [\mathcal{R}'(K)^* \mathcal{R}(K)] (K_e - K_i) \\ & \left([\mathcal{R}'(K)^* \mathcal{R}(K)] (K_e - K_i) \right) ds(\mathbf{x}) \end{aligned} \quad (34)$$

which is always ≤ 0 , such that we have in fact found a descent flow for the least squares cost functional $\mathcal{J}$.

Notice that (28) suggests immediately an alternative choice for the forcing term f , which is applicable also for the situation that some points at the boundary ∂D do not satisfy the requirement $|\nabla \psi| \neq 0$. Using the fact that formally $\delta(\psi) > 0$, we can define the new search direction as

$$f_d(\mathbf{x}) = -(K_e - K_i) \chi_{\psi,d}(\mathbf{x}) \mathcal{R}'(K)^* \mathcal{R}(K) \quad \text{for all } \mathbf{x} \in \Omega \quad (35)$$

where $\chi_{\psi,d}(\mathbf{x})$ is an arbitrary positive-valued approximation to $\delta(\psi)$ where the subscript d indicates the degree of approximation. In our numerical experiments we will use

$$\chi_{\psi,d}(\mathbf{x}) = \begin{cases} 1 & , \quad \text{there exists } \mathbf{x}_0 \in \Omega \text{ with } |\mathbf{x} - \mathbf{x}_0| < d \text{ and } \psi(\mathbf{x}_0) = 0 \\ 0 & , \quad \text{otherwise} \end{cases}$$

which we call the ‘narrow-band function’. Other approximations can be found for example in [17]. This search direction f_d , which is easy and stable to calculate, plugged into (28), as well gives us a descent flow for $\mathcal{J}(K)$.

Numerically, discretizing (35) by a straightforward finite difference time-discretization with time-step $\tau > 0$ yields at time t the update rule

$$\frac{\psi(t + \tau) - \psi(t)}{\tau} = (K_i - K_e) \chi_{\psi,d} \mathcal{R}'(K)^* \mathcal{R}(K). \quad (36)$$

Interpreting $\psi^{(n+1)} = \psi(t + \tau)$ and $\psi^{(n)} = \psi(t)$, we arrive at the iteration

$$\psi^{(n+1)} = \psi^{(n)} + \tau \delta \psi^{(n)}, \quad \psi^{(0)} = \psi_0, \quad (37)$$

with

$$\delta \psi^{(n)} = (K_i - K_e) \chi_{\psi^{(n)},d} \mathcal{R}'(K)^* \mathcal{R}(K) \quad \text{for all } \mathbf{x} \in \Omega. \quad (38)$$

Notice that the above update rule is still somewhat unsatisfactory since f_d and therefore $\delta \psi^{(n)}$ might be highly irregular. However, we want our level set function to have certain regularity properties, for example being Lipschitz-continuous. In order to alleviate this problem, we will assume now that $\psi \in H_1(\Omega)$ where

$$H_1(\Omega) = \left\{ \psi : \psi \in L_2(\Omega), \nabla \psi \in L_2(\Omega), \frac{\partial \psi}{\partial \nu} = 0 \text{ at } \partial \Omega \right\}. \quad (39)$$

Using this function space, we formally need to replace the adjoint operator $\mathcal{R}'(\psi)^*$ by a new adjoint operator $\mathcal{R}'(\psi)^\circ$ which maps back from the data space into this Sobolev space $H_1(\Omega)$. Using the weighted inner product

$$\langle v, w \rangle_{H_1(\Omega)} = \alpha \langle v, w \rangle_{L_2(\Omega)} + \beta \langle \nabla v, \nabla w \rangle_{L_2(\Omega)} \quad (40)$$

where $\alpha \geq 1$ and $\beta > 0$ are carefully chosen regularization parameters, and repeating the above derivation with this new function space, yields the regularized forcing term

$$\begin{aligned} f_r &= (\alpha I - \beta \Delta)^{-1} f_d \\ &= (\alpha I - \beta \Delta)^{-1} (K_i - K_e) \chi_{\psi,d} \mathcal{R}'(K)^* \mathcal{R}(K) \quad \text{for all } \mathbf{x} \in \Omega. \end{aligned} \quad (41)$$

The positive definite operator $(\alpha I - \beta \Delta)^{-1}$ has a smoothing effect on f_d since it maps from $L_2(\Omega)$ towards the smoother Sobolev space $H_1(\Omega)$. Different choices of the weighting parameters α and β visually have the effect of ‘smearing out’ the unregularized updates to a different degree. In particular, high frequency oscillations or discontinuities of the updates for the level set function are removed, which yields shapes with more regular boundaries. Notice that f_r is defined on the whole domain Ω , such that no extension velocity needs to be determined anymore when applying this regularization scheme. f_r will be the flow which we use in our numerical experiments. More information on the regularization scheme with the operator $(\alpha I - \beta \Delta)^{-1}$ can be found in [10], where it has been discussed in detail for a pixel-based reconstruction scheme.

6 The Adjoint Technique for Calculating Sensitivities

In this Section we show how to apply the adjoint linearized residual operator $\mathcal{R}'(K)^*$ to an arbitrary vector ρ of the data space Z in an efficient way. This

leads to the so-called ‘adjoint scheme’ for calculating sensitivities. For earlier work regarding the adjoint scheme in reservoir characterization we refer to, e.g., [6, 10, 13] and references given there.

6.1 Computation of the Operator $\mathcal{R}'(K)$. The Linearized Problem.

Let us consider a small perturbation δK in the permeability distribution K that leads to small perturbations W and q in the saturation and the pressure, respectively. Here we assume that the pressure remains nearly unchanged so that ∇q is negligible. This is so because the pressure is a smooth function compared to the saturation. Using a heuristic approach to derive an expression for $\mathcal{R}'$, we introduce $K + \delta K$ and $S_w + W$ in (7) and we neglect second order terms. Then, W solves the initial value problem

$$\begin{aligned} \phi \frac{\partial W}{\partial t} - \nabla \cdot \left[\frac{\partial T_w}{\partial S_w} W \nabla p \right] - \frac{\partial Q_w}{\partial S_w} W &= \frac{\delta K}{K} Q_w + \nabla \cdot \left[\frac{\delta K}{K} T_w \nabla p \right] \quad \text{in } \Omega \\ W(\mathbf{x}, 0) &= 0 \quad \text{in } \Omega \end{aligned} \quad (42)$$

where S_w and p are the solutions of (6)-(10). From the value of W we derive the linearized response of the data to a perturbation δK in the permeability distribution, which is given by

$$\mathcal{R}'(K)\delta K = \frac{\partial Q_w}{\partial S_w} W \Big|_{\Omega_+ \times [0, t_f]} . \quad (43)$$

Here, Ω_+ denotes the set of our production wells where data are collected.

6.2 Computation of the Operator $\mathcal{R}'(K)^*$. The Adjoint Problem.

Here, we derive an expression for the adjoint operator $\mathcal{R}'(K)^*$ applied to a function ρ in the data space. The operator $\mathcal{R}'(K)^*$ is defined by

$$\langle \mathcal{R}'(K)\delta K, \rho \rangle_Z = \langle \delta K, \mathcal{R}'(K)^*\rho \rangle_P \quad (44)$$

where $\langle \cdot, \cdot \rangle_P$ denotes the inner product in the parameter space P , and $\langle \cdot, \cdot \rangle_Z$ denotes the inner product in the data space Z . We define these inner products to be

$$\langle f, g \rangle_Z = \sum_{j=1}^{N_p} \int_0^{t_f} f_j g_j dt ; \quad \langle A, B \rangle_P = \int_{\Omega} A B d\mathbf{x} , \quad (45)$$

where $f_j = f(\mathbf{x}_{p_j}, t)$ and $g_j = g(\mathbf{x}_{p_j}, t)$, $j = 1, \dots, N_p$, are time-dependent functions defined at the production well positions $\mathbf{x}_{p_j}$. The following *adjoint form of the linearized residual operator* has been derived in [10]:

Let $\rho \in D$ be an arbitrary function in the data space. Then $\mathcal{R}'(K)^* \rho$ is given by

$$\mathcal{R}'(K)^* \rho = \int_0^{t_f} \left(\frac{T_w}{K} \nabla p \nabla z - z \frac{1}{K} Q_w \right) dt \quad (46)$$

where z is the solution of the adjoint equation

$$-\phi \frac{\partial z}{\partial t} + \frac{\partial T_w}{\partial S_w} \nabla p \nabla z - \left(z - \sum_{j=1}^{N_p} \rho \delta(\mathbf{x} - \mathbf{x}_j^{(p)}) \right) \frac{\partial Q_w}{\partial S_w} = 0 \quad \text{in } \Omega \quad (47)$$

$$z(\mathbf{x}, t_f) = 0 \quad \text{in } \Omega, \quad (48)$$

and S_w and p are the solutions of (6)-(10).

Notice that Q_w is nonzero only at the well locations. Therefore, when we assume in the mathematical derivation of the theorem that the permeability is known directly at the wells (these values are available from well-log data), the second term in (46) disappears and we only have to evaluate the first term in order to calculate the update in the rest of the domain Ω . This will be the approach we use in our numerical reconstructions.

7 The Algorithm

Let us assume that we are given some true data $\tilde{g}$ which either have been measured physically in a field experiment or (in our case) have been generated by running an independent Streamline simulator on the true geological setup. As already mentioned, both the Streamline simulator for generating data as well as the finite differences simulator IMPES (for the forward and the adjoint simulation) which are used during the reconstruction process have been written and implemented by our research group Modelling and Numerical Simulation at University Carlos III in Madrid. Using an independent Streamline simulator for generating data which is different from the simulator employed during the inversion process makes the inversion more realistic and avoids the so-called ‘inverse crime’. As in real physical experiments, the data calculated with the Streamline simulator will be sufficiently different from the data calculated by the IMPES simulator even when using the same correct physical model for both. That is due to the different ways how these two numerical schemes are constructed. Therefore, one of them can, for the purpose of testing the inversion algorithm in a controlled way, ‘play’ the role of the physical experiment until real physical data become available. The typical difference in the data (i.e., the ‘noise level’) calculated with the streamline method and with the IMPES method is about 3% [10].

Then, the resulting iterative algorithm for shape reconstruction using our level set method can roughly be summarized as follows:

- 1.) Compute the initial level set function $\psi^{(0)}$ as a signed distance function of an initial shape which takes into account the available prior information of the geological site. This a-priori information typically is available from well-log data which specify good approximate permeability values in a small neighborhood of the injector and producer wells. During the succeeding iterations, these permeability values directly at the wells (only one pixel per well) will be fixed. Put $n = 0$.
- 2.) Use the IMPES simulator for solving the forward problem (6)–(10) on the latest best guess $K(\psi^{(n)})$ for the geological shapes. The corresponding shape is $D^{(n)}$ with boundary $\Gamma^{(n)}$. This yields the measurement vector $g^{(n)}$. Calculate the residuals $\mathcal{R}(\psi^{(n)}) = g^{(n)} - \tilde{g}$.
- 3.) Solve the adjoint problem (47), (48) with $\rho = \mathcal{R}(\psi^{(n)})$. Then $\delta K = \mathcal{R}'(K)^* \rho$ is given by equation (46).
- 4.) Build the narrowband $\chi_{\psi^{(n)},d}$ which is a matrix with ones in a small neighborhood of $\Gamma^{(n)}$ and zeros elsewhere. Calculate

$$\delta\psi^{(n)} = (K_i - K_e) \chi_{\psi^{(n)},d} \delta K \quad (49)$$

according to (35).

- 5.) Compute

$$\widehat{\delta\psi}^{(n)} = (\alpha I - \beta \Delta)^{-1} \delta\psi^{(n)} \quad (50)$$

with prescribed regularization parameters $\alpha > 0$ and $\beta > 0$. This yields the regularized update $\widehat{\delta\psi}^{(n)}$

- 6.) Apply the update

$$\psi^{(n+1)} = \psi^{(n)} + \tau^{(n)} \widehat{\delta\psi}^{(n)} \quad (51)$$

where the step-size $\tau^{(n)}$ is determined by some line-search criterion. (In our numerical experiments, this will be a pre-specified maximal number of pixels which change value in a given step). Rescale the result such that the minimum of the level set function has a fixed value.

- 7.) Verify stopping criterion. If not reached yet, go back to step 2.) with $n = n + 1$ and continue from there, using the now updated shape and level set function for finding a new correction. Continue this procedure in an iterative manner.

8 Numerical Examples

In order to verify the performance of our shape based inversion algorithm using level sets for realistic examples we have investigated two test cases in

two spatial dimensions. These will be discussed in the following. In both cases, the dimensions of the reservoir are 600 m by 600 m discretized into 25×25 grid cells. There are 9 production wells and 4 injection wells arranged as an array of 4 so-called five-spot patterns (one five-spot pattern consists here of one injection well surrounded by 4 production wells) (see Figure 1). It is assumed to be known that the reservoir consists in average of two lithofacies, namely shale and sand. The permeability of shale is given as 250 milli-Darcy (mD), and that of sand as 1500 mD. This means, there is a high contrast between these two regions. All physical parameters of the reservoir are assumed to be (approximately) known except of the topology of these two regions, which needs to be reconstructed from the data. There are two (incompressible) fluids in the reservoir, water and oil. In our numerical simulator, we use tabulated values for the relative permeabilities K_{rw} and K_{ro} as shown in [10], which correspond to a Corey function with coefficients $n_w = 3$ and $n_o = 2$. The viscosity values for oil and water are $\mu_o = 0.79 \times 10^{-3}$ Pa s and $\mu_w = 0.82 \times 10^{-3}$ Pa s, and the porosity is taken to be constant $\phi = 0.213$ in the reservoir. The pressure values in the reservoir are in the range between 2000 psi (imposed pressure at production wells) and 3500 psi (imposed pressure at injection wells). The numerical physical time-step (which is unrelated to the time-step of the artificial shape evolution) used in the simulator is 0.1 days, and the reservoir is monitored over a period of 120 days. For more details regarding our reservoir simulation tools, we refer again to [10].

8.1 First Test Case

The first numerical example describes a situation of five regions which are arranged in an alternating banded fashion. See Figure 1. The central band as well as the corner bands consist of shale, whereas the two other bands consist of sand. The initial guess is displayed in Figure 1. The initial level set function is the corresponding signed distance function which is calculated analytically. Starting with this initial guess, the algorithm as described in Section 7 calculates repeated updates for the level set function with the line search criterion that in each step not more than 4 pixels change their permeability values. This criterion has mainly been chosen in order to arrive at a smooth evolution. In practice, much bigger steps can be taken using different line-search criteria for the cost of a less smooth evolution. The width of the narrowband is 3 pixels. The regularization parameters α and β have been chosen according to criteria explained in [10]. No specific stopping criterion is applied here in order to monitor the general behavior of the algorithm. We stop the algorithm when the cost does not change anymore significantly or when a maximal number of iterations has been reached. Figure 1 shows the true shape (top left), the initial guess (top right), the final reconstruction after 150 iterations (bottom left) and the evolution of the least squares norm of the data misfit (bottom right). The evolution of the shape during the reconstruction is displayed in Figure 2. In addition, in the bottom right image of this figure we show the to-

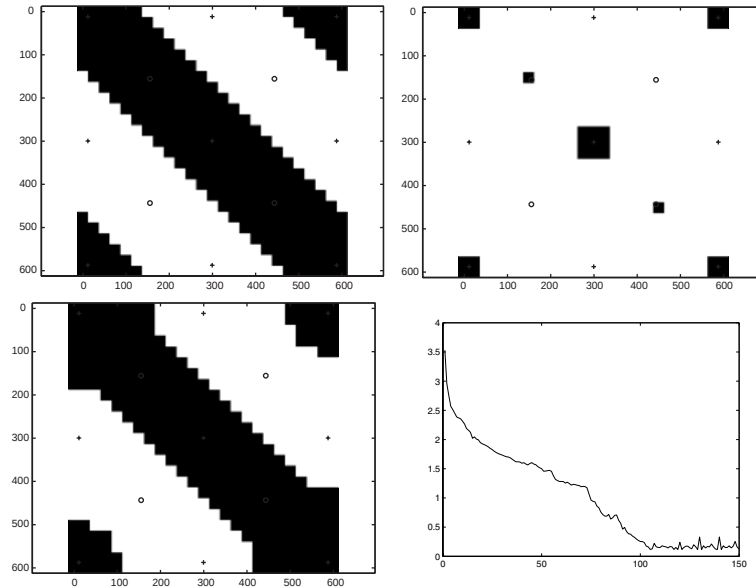


Fig. 1. First example, first initial guess. Top left: true shape; top right: initial guess; bottom left: final reconstruction after 150 iterations; bottom right: evolution of least squares data misfit. Injection wells are indicated by circles ($\circ$), production wells by pluses ($+$).

tal water production rate (in m^3/s) for the initial model, the reference model and the final reconstruction as an alternative way of verifying the corresponding match with the production data. It can be clearly seen that topological changes occur during the evolution and are handled without problems by the algorithm.

In the above reconstruction the initial guess assumes as starting point for the evolution that the reservoir consists of sand at those points where no prior information is available (i.e., at some distance away from the wells). An interesting question is whether the final reconstruction would change significantly if we assume in our initial guess that these points consist of shale. In order to investigate this question, we have run one additional numerical experiment for this situation using the initial guess as displayed in the top right image of Figure 3. Figures 3 and 4 show the results of the corresponding evolution. We observe that the final reconstruction does not change significantly when using this alternative choice for the initial guess.

8.2 Second Test Case

The second numerical example is similar to a situation which has been considered in [10] using a pixel-based reconstruction scheme. It consists of a central sand region of high permeability surrounded by shale (see Figure 5). Again,

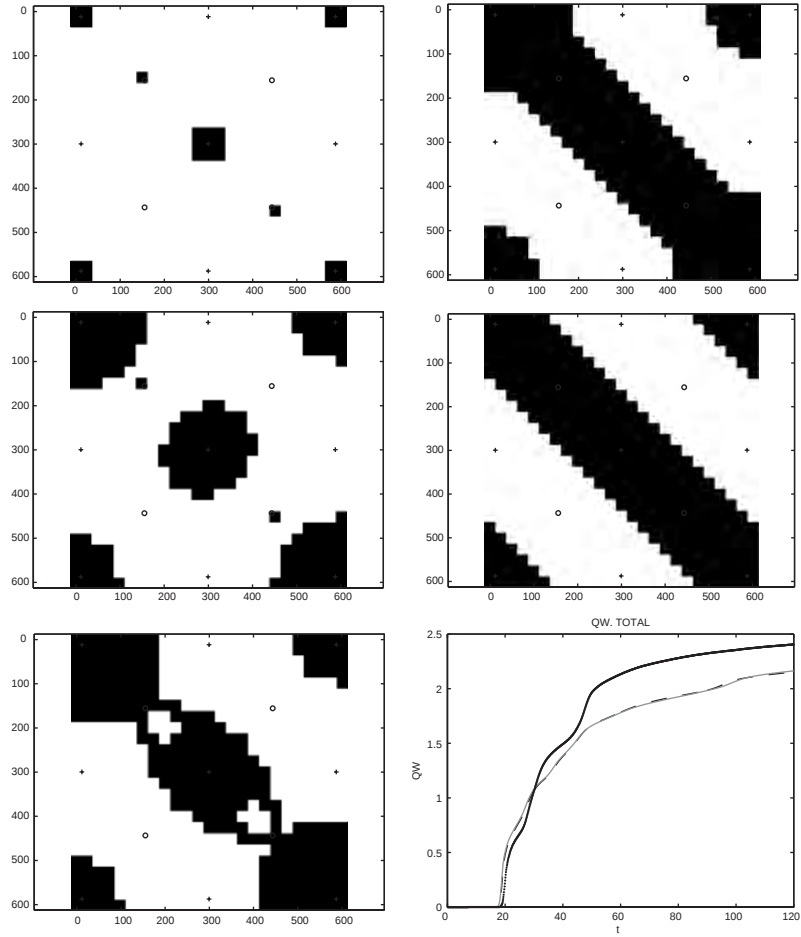


Fig. 2. Shape evolution for first example, first initial guess. Left column from top to bottom: initial guess, after 40 and 80 iterations; right column from top to bottom: after 150 iterations and reference model; Bottom right image: total water production rate in m^3/s versus days for initial model (upper curve), the reference model (lower solid) and the final reconstruction (lower dashed).

the initial level set function is a signed distance function of the initial guess for the shape. In this example, we use the line search criterion that in each step not more than 5 pixels change their permeability values. The width of the narrowband is 3 pixels. Figure 5 shows the true shape (top left), the initial guess (top right), the final reconstruction after 200 iterations (bottom left) and the evolution of the least squares norm of the data misfit (bottom right). The evolution of the shape during the reconstruction is displayed in Figure 6. Also here, topological changes occur during the evolution and are handled without problems by the algorithm.

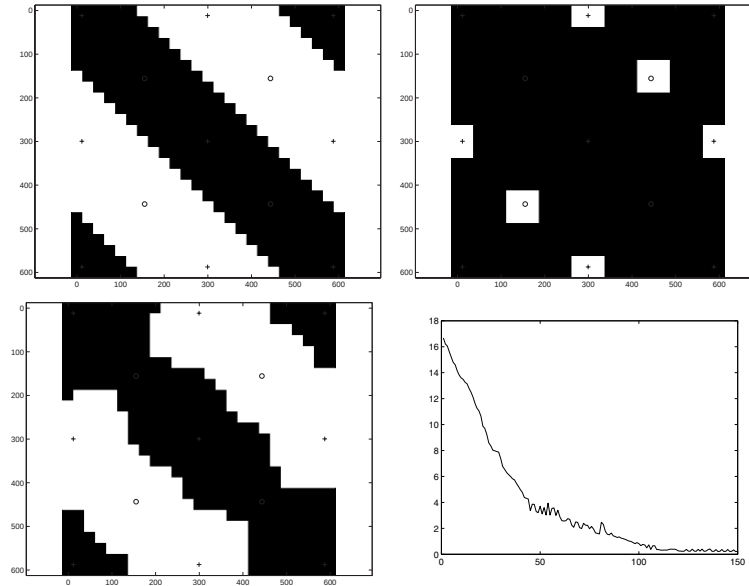


Fig. 3. First example, second initial guess. Top left: true shape; top right: initial guess; bottom left: final reconstruction after 150 iterations; bottom right: evolution of least squares data misfit. Injection wells are indicated by circles ($\circ$), production wells by pluses ($+$).

9 Conclusions and Future Work

We have presented a new algorithm for shape reconstruction from two-phase incompressible flow data in the application of history matching in reservoir characterization. The method uses a level set representation for the shapes during the artificial evolution which is designed to minimize the least squares misfit between calculated and real data. Moreover, an adjoint scheme is employed for calculating shape sensitivities in an efficient way. A flexible regularization tool has been introduced into this shape reconstruction approach which stabilizes the inversion and which yields relatively smooth boundaries in the reconstructed shapes. We have presented two numerical examples for realistic situations which show that the method is able to reconstruct quite complicated shapes from relatively few production data which have been generated by a different reservoir simulator. Topological changes which occur during the artificial shape evolution are handled easily and in an automatic way by the level set formulation.

In our future work, we plan to compare the performance of this algorithm to alternative shape-based reconstruction schemes. Moreover, we want to extend the algorithm to simultaneously reconstructing the shapes and the two parameter values inside the different zones (or even more detailed information regarding these zones) from the given data. The incorporation of additional

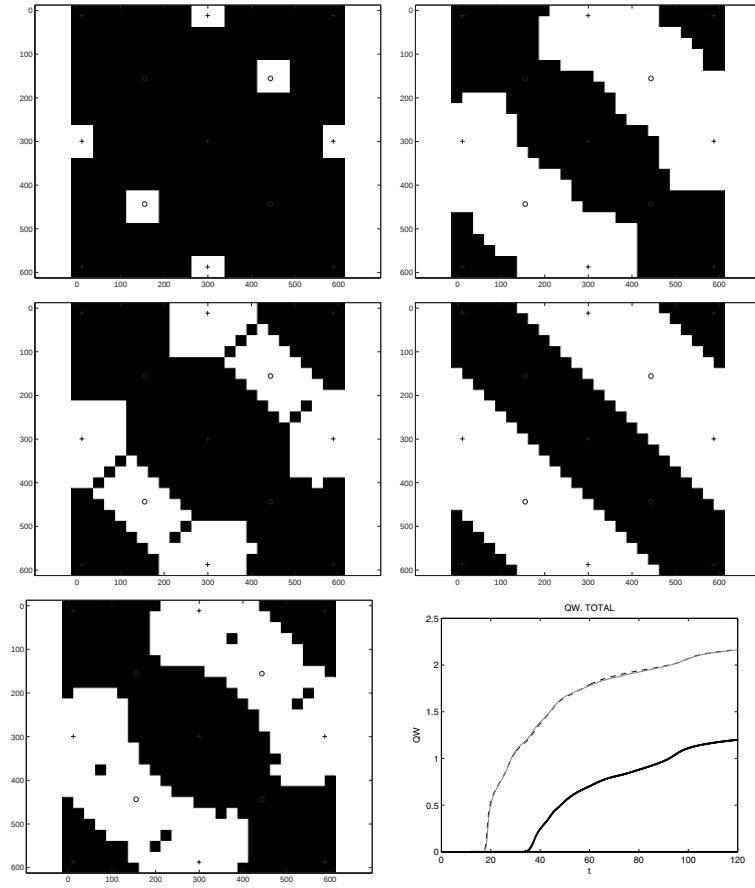


Fig. 4. Shape evolution for first example, second initial guess. Left column from top to bottom: initial guess, after 50 and 100 iterations; right column from top to bottom: after 150 iterations and reference model; Bottom right image: total water production rate in m^3/s versus days for initial model (lower curve), the reference model (upper solid) and the final reconstruction (upper dashed)

prior information (making use of e.g., statistical reservoir characterization techniques) is another interesting research topic which we would like to address in our future work. Finally, the ultimate goal will be to implement the level set based shape reconstruction algorithm in a more realistic 3D setup, such that it can be applied directly to real production data provided by reservoir engineers.

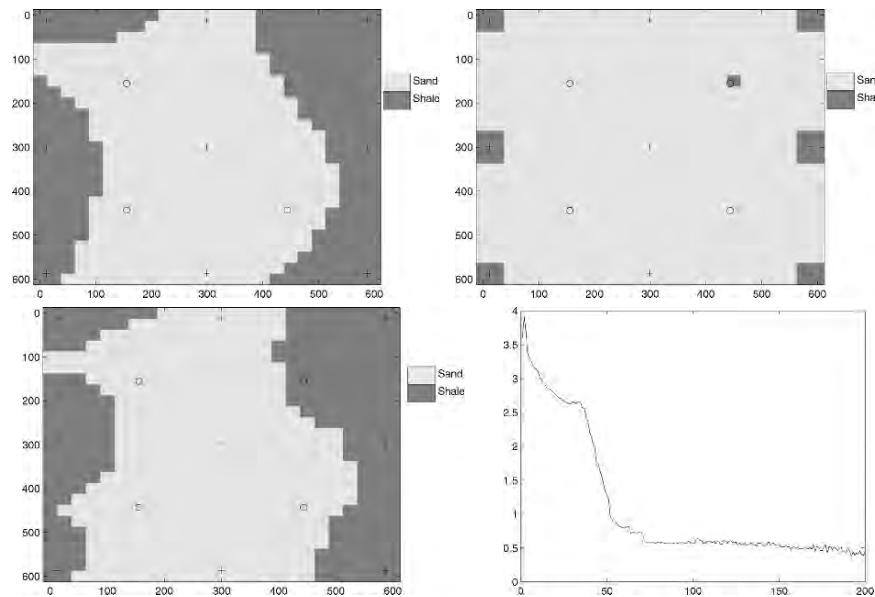


Fig. 5. Second example. Top left: true shape; top right: initial guess; bottom left: final reconstruction; bottom right: evolution of least squares data misfit. Injection wells are indicated by circles (o), production wells by pluses (+).

Acknowledgments

Funding for this work was provided by the Dirección de Tecnología y Soporte Técnico, Repsol-YPF.

References

1. Agarwal B and Blunt M 2003, A streamline-based method for assisted history matching applied to an Arabian Gulf field, SPE paper 84462, SPE ATCE, Denver, CO, Oct. 5-8.
2. Berre I, Lien M and Mannseth T 2005, A level set corrector to an adaptive multiscale permeability prediction, *Preprint*.
3. Bourdarot G 1998 Well Testing: Interpretation Methods, Editions Technip.
4. Carter R D, Kemp Jr L F, Pierce A C and Williams D L 1974 Performance Matching With Constraints *SPE Journal* **14** 187-196.
5. Chan T F and Tai X-C 2003 Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients *J. Comput. Physics* **193** pp. 40-66.
6. Chavent G, Dupuy M and Lemmonier C 1975 History Matching by Use of Optimal Control Theory *SPE Journal* **15** 74-86.
7. Crichlow H B 1976 Modern Reservoir Engineering-A Simulation Approach, Prentice Hall, New Jersey.

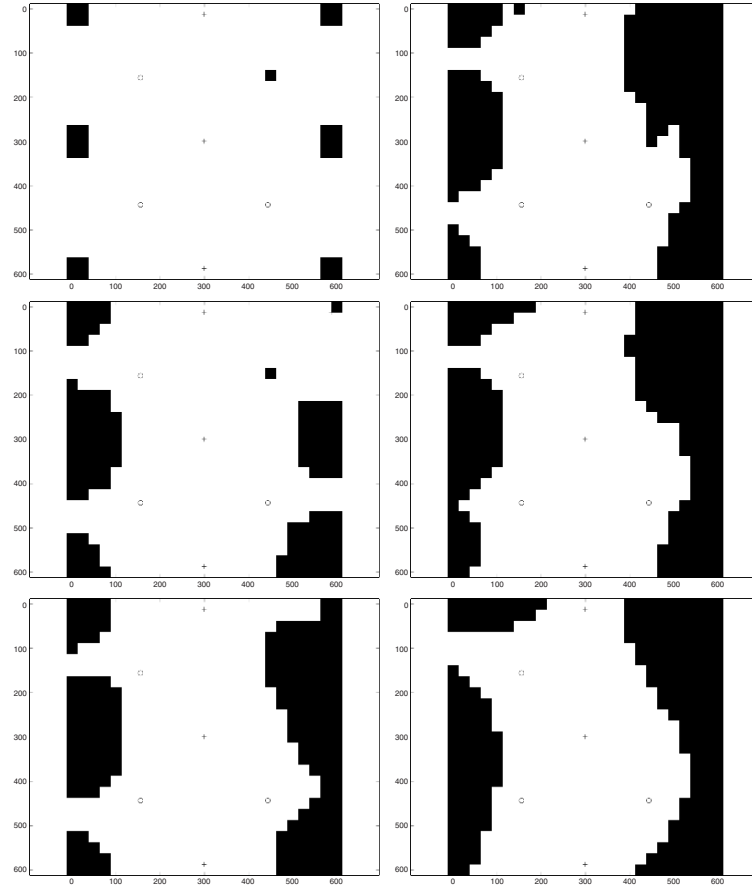


Fig. 6. Shape evolution for second example. Left column from top to bottom: initial guess, after 25 and 50 iterations; right column from top to bottom: after 150 and 200 iterations; bottom right image: true shape.

8. Dorn O, Miller E L and Rappaport C 2000, A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets, *Inverse Problems* **16**, 1119–1156.
9. Ewing R E, Pilant M S, Wade J G and Watson A T 1995 Identification and Control Problems in Petroleum and Groundwater Modeling, *Control Problems in Industry* (I. Lasciecka and B. Morton, eds.), Progress in Systems and Control Theory, Birkhauser, 119–149.
10. González-Rodríguez P, Kindelan M, Moscoso M and Dorn O, *History matching problem in reservoir engineering using the propagation back-propagation method*, *Inverse Problems* **21**, 565–590, 2005.
11. Karlsen K H, Lie K-A and Risebro N H, A fast marching method for reservoir simulation, *Comput. Geosci.*, **4**, 185–206, 2000.

12. Jacquard P and Jain C 1965 Permeability Distribution From Field Pressure Data *SPE Journal* **5** 281-294.
13. Li R, Reynolds A C and Oliver D S 2003 History matching of three-phase flow production data, *SPE Journal* **8** 328-340.
14. Lien M, Berre I and Mannseth T 2005 Combined adaptive multiscale and level set parameter estimation, *Preprint*.
15. Litman A, Lesselier D and Santosa F, *Reconstruction of a two-dimensional binary obstacle by controlled evolution of a level-set*, *Inverse Problems* **14**, 685-706, 1998.
16. Osher S and Shethian J, *Fronts propagation with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations*, *J. Comput. Physics.* **56**, 12-49, 1988.
17. Osher S and Fedkiw R, *Level set methods and dynamic implicit surfaces*, Springer, New York, 2003.
18. Rahon D, Edoa P F and Masmoudi M, *Inversion of Geological Shapes in Reservoir Engineering Using Well-Test and History Matching Production Data*, SPE 38656, 1997.
19. Rahon D, Edoa P F, and Masmoudi M, *Identification of Geological Shapes in Reservoir Engineering by History Matching Production Data*, SPE 48969, 1998.
20. Santosa F 1996 A level-set approach for inverse problems involving obstacles, *ESAIM: Control, Optimization and Calculus of Variations* **1**, 17-22.
21. Sethian J A, *Level Set Methods and Fast Marching Methods*, (2nd ed) Cambridge University Press, 1999.
22. Sun N and Yeh W G 1990 Coupled inverse problems in groundwater modelling 1. Sensitivity analysis and parameter identification *Water resources research* **26**, 2507-2525.
23. Vasco D M Yoon S and Datta-Gupta A 1999 Integrating Dynamic Data Into High-Resolution Reservoir Models Using Streamline-Based Analytic Sensitivity Coefficients *SPE Journal* **4** 389-399.
24. Wasserman M L, Emanuel A S and Seinfeld J H 1975 Practical Applications of Optimal-Control Theory to History-Matching Multiphase Simulator Models *SPE Journal* **15** 347-355.

Reservoir Description Using a Binary Level Set Approach with Additional Prior Information About the Reservoir Model

Lars Kristian Nielsen, Xue-Cheng Tai, Sigurd Ivar Aanonsen, and Magne S. Espedal

Department of Mathematics, University of Bergen and CIPR-Centre for Integrated Petroleum Research, University of Bergen. E-mail: {larskn,tai}@mi.uib.no, Sigurd.Aanonsen@cipr.uib.no, and resme@mi.uib.no

Summary. This paper considers the inverse problem of estimating the permeability for porous media flow. In the parameter estimation process we utilise data from the wells (production data) and spatially distributed data (from time-lapse seismic data), and in addition there are abilities to incorporate prior information about the sought solution. The closeness of the estimated model to the prior model and the fit of the simulated data to the measurements are measured in one single objective function.

In the solution process we approximate the permeability field by a piecewise constant function, and allow the discontinuity curves to have arbitrary shape with some forced regularity. To achieve this, we have utilised level set functions to represent the permeability field and applied an additional total variation regularisation.

The level set method of choice is a binary level set formulation which has the ability to both determine the curves of discontinuities and the constant values for each region. To solve the optimisation problem we have applied a variational augmented Lagrangian approach.

Key words: Inverse problems, reservoir description, parameter identification, two-phase flow, level set methods, augmented Lagrangian optimisation, total variation regularisation.

1 Introduction

Conservation of mass for two-phase, incompressible, immiscible, horizontal flow in a porous medium with isotropic permeability gives;

$$\Phi(\mathbf{x}) \frac{\partial S_o}{\partial t} - \nabla \cdot \left(\frac{\kappa(\mathbf{x}) \kappa_{ro}(S_o)}{\mu_o} \nabla p_o \right) = f_o(\mathbf{x}) , \quad (1)$$

$$\Phi(\mathbf{x}) \frac{\partial S_w}{\partial t} - \nabla \cdot \left(\frac{\kappa(\mathbf{x}) \kappa_{rw}(S_w)}{\mu_w} \nabla p_w \right) = f_w(\mathbf{x}) , \quad (2)$$

where $(\mathbf{x}, t) \in \Omega \times [0, T]$. $\Omega \in \mathbb{R}^2$ is a bounded reservoir domain and the subscripts o and w refer to the phases, water and oil, respectively. S_l denotes the saturation, μ_l the viscosity, p_l the pressure, f_l the external volumetric flow rate and κ_{rl} is the relative permeability, where l is the fluid phase. The porosity and the absolute permeability are given by $\Phi(\mathbf{x})$ and $\kappa(\mathbf{x})$, respectively.

Closure of the system is obtained through an assumption of a completely saturated medium,

$$S_o + S_w = 1, \quad (3)$$

and a supposed known function P_c defining the capillary pressure,

$$p_o - p_w = P_c(S_w). \quad (4)$$

The quantities Φ , κ , κ_{ri} and P_c are all dependent of the porous medium and are not accessible through direct measurements.

The problem treated in this paper is to find an estimate of the absolute permeability, $\kappa(\mathbf{x})$, when Φ and κ_{rl} are assumed to be known, and P_c is set to zero. To recover the permeability trends we will utilise information from the wells together with seismic data. This data does not give any direct information of the permeability, but through the equations of fluid flow, (1) - (4), we can use this indirect information to estimate the permeability on a coarse scale. A problem of this kind is generally known as an inverse problem, or more specific referred to as a *history-matching* problem.

It is well known that inverse problems often are ill-conditioned. Even though the data in this case is distributed both in time and space, the total distribution of data may still be sparse, see [25]. A sparse distribution of the data will usually make the conditioning of the problem worse [12]. To improve the conditioning of the problem it has to be regularised in a proper way. This can be done by restricting the parameter space in order to exclude non-physical solutions.

As in [25] we will use a level set method to represent the permeability. The level set method will force the solution to be piecewise constant. The geometries of the discontinuity curves are allowed to be arbitrary, but with some forced regularity achieved by a total variation regularisation.

Level set methods can produce piecewise constant solutions with a predefined number of constant levels. If it is natural to represent the sought solution with a fewer number of regions than this predefined number, the estimate will leave one or more regions empty. In this way we only need an upper bound for the number of regions in the piecewise constant solution.

The original level set method was proposed by Osher and Sethian [26] for tracing interfaces between different phases of fluid flow. It has later been a versatile tool for representing and tracking interfaces separating a domain into subdomains. The method has been applied in a wide range of applications, i.e., inverse problems, image analysis and optimal shape design problems. For a recently survey of level set methods see [29]. Examples of level set methods applied to inverse problems can be found in [6, 8, 10, 15, 11, 27, 3, 4, 5, 7].

In this work, we shall apply a variant of a *piecewise constant level set method* [18, 19, 20, 24, 30]. In these methods the level set functions are discontinuous and have discontinuities at the boundaries of the subdomains. The method of choice is a *binary level set method*, where the level set functions are required to only take the values 1 and -1 at convergence. This method has previously been applied for segmentation of digital images [20] and for solving inverse elliptic problems [24]. In [25] the same framework was applied for solving the history matching problem.

In this work we take the approach from [25] further by incorporating prior information of the model in the optimisation process. This information can generally be knowledge about parameters defining the model, such as functions for P_c , κ_{rl} , κ , etc. In our case we intend to find the optimal κ , while the other parameters are assumed known, and we have therefore included prior information about the structures in κ . We will penalise deviations from the prior model by measuring the fit of the estimate and the prior model in the same objective function as we measure the fit of the measured and the simulated data.

A requirement for applying the level set method on this problem is that we have indications of a piecewise constant field. The geological permeability maps can contain such information, and therefore also information about what is a suitable bound for the number of constant levels. The method presented in the theory part of this paper is a multiple level set approach able to find an arbitrary number of regions. In the numerical part we will though restrict ourselves to look at fields where we assume there are indications of a channelled system with two different levels.

The paper is organised in the following way: In Section 2 the inverse problem is defined. The general framework for the binary level set approach is presented in Section 3, while we in Section 4 explain how this framework is utilised to solve the inverse problem. Further the numerical optimisation method and the applied algorithm are given in Section 5, and the numerical results are presented in Section 6. Finally, a summary and the conclusions are given in Section 7.

2 The Inverse Problem

Because of the nature of the permeability it is more natural to solve the optimisation problem with respect to the logarithm of the permeability instead of the permeability itself. For notational matter we define

$$q(\mathbf{x}) = \log_{10} \kappa(\mathbf{x}) , \quad (5)$$

and solve the problem with respect to $q(\mathbf{x})$. The transformation from κ to q will only influence the jumps between the different permeability zones, and not the contour of the discontinuities. This is because a piecewise constant κ is

equivalent to a piecewise constant q . When obtaining a solution, the estimate of $q(\mathbf{x})$ can easily be transformed back to the permeability $\kappa(\mathbf{x})$ through (5).

Let $\mathbf{d}_{\text{well}}$ be a vector of well data, and $\mathbf{d}_{\text{seis}}$ be a vector of seismic data, and assume that all measurements have been transformed into pressures and saturations:

$$\begin{aligned}\mathbf{d}_{\text{well}} &= \{p_o(\mathbf{x}_{\text{well},i}, t), S_w(\mathbf{x}_{\text{well},i}, t) \quad \text{for } i = 1, 2, \dots, n_{\text{well}}, t \in [0, T]\} , \\ \mathbf{d}_{\text{seis}} &= \{p_o(\mathbf{x}, t_j), S_w(\mathbf{x}, t_j) \quad \text{for } \mathbf{x} \in \Omega, j = 1, 2, \dots, n_{\text{seis}}\} ,\end{aligned}$$

where n_{well} is the number of present wells in Ω and n_{seis} is the number of seismic surveys in the time domain $[0, T]$. A conversion of seismic response to pressure and saturation is difficult. However, this topic is under research; see, for example, [14, 16, 22].

There are several alternatives in which the mismatch between simulated and measured data can be performed. An alternative approach to the presented solution method could be to transform the simulated values of p_o and S_w to elastic parameters by a petro-elastic model (see, e.g., [14]). Furthermore, corresponding elastic parameters could be inverted from seismic measurements (see, e.g., [31]), and the mismatch could be performed using elastic parameters as performed in [1]. We believe that the alternative of mismatch should not influence the evaluation of the solution approach presented in this paper, and have for simplicity measured the mismatch on values of p_o and S_w .

Furthermore, we shall assume we have some prior information about the model. In our case we suppose we have a geological model which contains information about the shapes of the channels or barriers in the reservoir. That is, we know something about the high and low permeability zones in the reservoir, and can construct an initial model, q_{prior} , of the permeability field.

When incorporating different kinds of data in one optimisation process, it is important to weight the different data types properly. Following the approach from [2, 1, 25] we apply the following objective function to measure the error in the solution;

$$\begin{aligned}J_{\text{tot}}(q) &= J_{\text{well}}(q) + J_{\text{seis}}(q) + J_{\text{prior}}(q) \\ &= \frac{1}{2}(\mathbf{d}_{\text{well}} - \mathbf{m}_{\text{well}}(q))^T D_{\text{well}}^{-1}(\mathbf{d}_{\text{well}} - \mathbf{m}_{\text{well}}(q)) \\ &\quad + \frac{1}{2}(\mathbf{d}_{\text{seis}} - \mathbf{m}_{\text{seis}}(q))^T D_{\text{seis}}^{-1}(\mathbf{d}_{\text{seis}} - \mathbf{m}_{\text{seis}}(q)) \\ &\quad + \frac{1}{2}(q - q_{\text{prior}})^T D_{\text{prior}}^{-1}(q - q_{\text{prior}}) .\end{aligned}\tag{6}$$

Here $\mathbf{m}_{\text{well}}(q)$ and $\mathbf{m}_{\text{seis}}(q)$ are the simulated values corresponding to the given measurements. These values are calculated by the forward model ((1)-(4)) for a given function $q(\mathbf{x})$ (or corresponding permeability function $\kappa(\mathbf{x})$). The estimate in (6) is corresponding to a minimum-variance estimate where the norm of the solution error is minimised; see, e.g., [21]. Neglecting the model

error, the elements in the covariance matrices, D_{well} , D_{seis} and D_{prior} , should hence represent uncertainties in the different measurements. The covariance matrices will in general not be diagonal, see [1]. The two first terms of the objective function, J_{well} and J_{seis} , will correspond to the misfit between the measured and the simulated data. Contrary to the objective function used in [25], we have in this work added an additional term, J_{prior} , which will penalise deviations from a known prior model.

The problem of recovering $q(\mathbf{x})$ is an inverse problem which can be highly ill-posed. Because of the ill-posedness, a proper regularisation is required to restrict the solution space. In this work we restrict the solution to be piecewise constant. We will allow for arbitrary shapes of the geometries of the discontinuity curves, but with some restrictions related to the regularity of q . As in [24, 8] this is achieved by applying a total variation based regularisation together with the piecewise constant requirement. The actual applied regularisation is

$$R(q) = \int_{\Omega} |\nabla q| d\mathbf{x} , \quad (7)$$

and this will both control the length of the interfaces and the jumps of q .

The functional to be minimised is defined as

$$F(q) = J_{\text{tot}}(q) + \beta R(q) , \quad (8)$$

where $\beta > 0$ is a parameter weighting the amount of regularisation. The inverse problem is solved by finding the optimal function q^* , which is the solution of the following minimisation problem:

$$q^* = \arg \min_{q \in Q} F(q) , \quad (9)$$

where Q is a space of piecewise constant functions.

3 The Binary Level Set Approach

The binary level set method is a type of piecewise constant level set methods where the discontinuities of the level set functions are defining the division of a domain into a number of subdomains. This is contrary to the traditional level set formulation [26], where the zero level set of a continuous level set function is utilised for this purpose.

Chan and Vese [9] proposed a level set framework for image segmentation using the Mumford-Shah model. This framework was taken further by Lie et al. [20] where the binary level set method was proposed for segmentation problems. Some of the essential ideas for this method have appeared earlier in [13, 28]. The binary level set method has later been used in [24] for solving inverse elliptic problems and in [25] for solving inverse two-phase flow problems. These works do not provide an explicit comparison of the binary and

the traditional level set formulation, but it has been shown that the binary method is able to solve the given problems in a satisfactory way.

The binary level set method provide an easy way to implement level set problems. Compared to the traditional formulation, both reinitialisation of the level set functions (required to force the continuous level set functions to be signed distance functions which is advantageous for numerical reasons) and problems related to non-differentiability of the Heaviside function are avoided. The discontinuities present by the Heaviside function are instead incorporated directly in the level set function. In this section we will present the framework applied in [24, 25], where the level set functions are utilised to construct a piecewise constant coefficient function as a solution to an inverse problem.

In the binary level set formulation, the level set functions are discontinuous function which at convergence should take the values -1 or 1. These functions can be used to partition a domain Ω into a number of subdomains $\{\Omega_j\}$ by requiring different sign of the level set functions inside and outside the subdomains. In this way, the discontinuities of the functions will represent the boundary of the subdomains.

Let us first assume that Ω need to be divided into two subdomains, Ω_1 and Ω_2 , such that $\Omega_1 \cap \Omega_2 = \emptyset$ and $\Omega = \bar{\Omega}_1 \cup \bar{\Omega}_2$, where $\bar{\Omega}_j$ is the closure of Ω_j . A representation of this domain can be given by

$$\phi(\mathbf{x}) = \begin{cases} 1 & \forall \mathbf{x} \in \Omega_1 \\ -1 & \forall \mathbf{x} \in \Omega_2, \end{cases} \quad (10)$$

and the curve separating Ω_1 and Ω_2 is implicitly given as the discontinuity of ϕ , see Figure 1. The properties of ϕ can be used to construct a scalar function $q(\mathbf{x})$ with distinct constant values inside the two different subdomains. If we assume that the value of $q(\mathbf{x})$ is equal to c_1 in Ω_1 and equal to c_2 in Ω_2 , then q may be written as

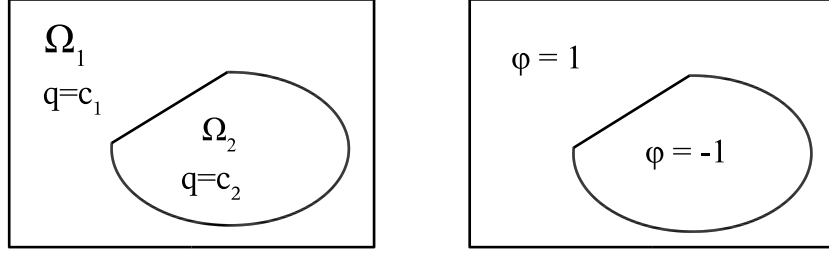
$$q = \frac{1}{2} [c_1(\phi + 1) - c_2(\phi - 1)] . \quad (11)$$

Multiple level set functions can be used to represent more than two regions. Following the terminology applied in [20], a function having four constant regions can be represented by two level set functions, and expressed as

$$q = \frac{1}{4} [c_1(\phi_1 + 1)(\phi_2 + 1) - c_2(\phi_1 + 1)(\phi_2 - 1) - c_3(\phi_1 - 1)(\phi_2 + 1) + c_4(\phi_1 - 1)(\phi_2 - 1)] . \quad (12)$$

Further, N binary level set functions can be combined to produce a coefficient function with 2^N different levels. Given $\boldsymbol{\phi} = \{\phi_i\}_{i=1}^N$ and $\mathbf{c} = (c_1, c_2, \dots, c_{2^N})$, the function q can be expressed as the sum

$$q(\boldsymbol{\phi}, \mathbf{c}) = \sum_{j=1}^{2^N} c_j \psi_j(\boldsymbol{\phi}) , \quad (13)$$



(a) The discontinuity of a piecewise constant function $q(\mathbf{x})$.

(b) Binary level set functions.

Fig. 1. A representations of a piecewise constant function $q(\mathbf{x})$. In this example q has two regions with different constant values, c_1 and c_2 . By binary level set functions the discontinuity of q can be represented as in Figure (b).

where ψ_j are basisfunctions dependent on ϕ . An expression for ψ_j is omitted here, but can be found in [24]. Equations (11) and (12) are special cases of (13). In the first case (i.e., (11)), we have $\psi_1 = \frac{1}{2}(\phi + 1)$ and $\psi_2 = -\frac{1}{2}(\phi - 1)$. With two level set functions, we get $\psi_1 = \frac{1}{4}(\phi_1 + 1)(\phi_2 + 1)$, $\psi_2 = -\frac{1}{4}(\phi_1 + 1)(\phi_2 - 1), \dots$ in (12).

In the following, we let $K(x) = x^2 - 1$. The level set functions are required to satisfy the constraint

$$K(\phi_i) = \phi_i^2 - 1 = 0 \quad \forall i. \quad (14)$$

This requirement will force the level set functions to take the values -1 or 1 at convergence. With (14) fulfilled, the basis functions will be characteristic functions for the corresponding subdomains, i.e., $\psi_j = 1$ in Ω_j and zero elsewhere. That is, the support of the different basis functions are non-overlapping, $(\text{supp } \psi_i) \cap (\text{supp } \psi_j) = \emptyset \quad \forall i \neq j$, and the total support of all the basisfunctions covers the complete domain, i.e. $\Omega = \cup_{j=1}^{2^N} (\text{supp } \psi_j)$.

4 The Binary Level Set Method for the Inverse Problem

From the last section, we see that every piecewise constant function can be represented as in (13) under the requirement that the level set functions satisfy (14). In order to find a piecewise constant function, we just need to find the corresponding c_j -values and the level set functions ϕ_i . If we define the vector $\mathbf{K}(\phi) = \{K(\phi_i)\}_{i=1}^N$, we can thus reformulate (9) as

$$(\phi^*, \mathbf{c}^*) = \arg \left\{ \min_{\phi, \mathbf{c}} F(q(\phi, \mathbf{c})) \right\} \quad \text{subject to} \quad \mathbf{K}(\phi) = \mathbf{0}, \quad (15)$$

where the optimal coefficient can be calculated by $q^* = q(\phi^*, \mathbf{c}^*)$. The constraint $\mathbf{K} = \mathbf{0}$ is applied to control the structure of the level set functions, and will therefore depend on the choice of basis functions.

Define $\tilde{F}(\phi, \mathbf{c}) = F(q(\phi, \mathbf{c}))$. To evolve the level set functions and update the constant values such that $q(\mathbf{x})$ will converge to the optimal solution, we need to calculate the derivatives of $\tilde{F}$ with respect to ϕ and $\mathbf{c}$. By the chain rule we have, c.f. [8],

$$\frac{\partial \tilde{F}}{\partial \phi_i} = \frac{\partial F}{\partial q} \frac{\partial q}{\partial \phi_i} \quad \forall i = 1, 2, \dots, N \quad (16)$$

and

$$\frac{\partial \tilde{F}}{\partial c_j} = \int_{\Omega} \frac{\partial F}{\partial q} \frac{\partial q}{\partial c_j} d\mathbf{x} \quad \forall j = 1, 2, \dots, 2^N. \quad (17)$$

The time consuming part of these calculations is to find $\frac{\partial F}{\partial q}$. In this work $\frac{\partial F}{\partial q}$ is calculated by adjoint gradient calculations (see, e.g., [17, 23]) in a reservoir simulator.

5 Numerical Optimisation

We apply an augmented Lagrangian method to solve (15) numerically. The Lagrangian functional involves both $\tilde{F}$ and the constraint K ;

$$L(\phi, \mathbf{c}, \boldsymbol{\lambda}) = \tilde{F}(\phi, \mathbf{c}) + \sum_{i=1}^N \int_{\Omega} \lambda_i K(\phi_i) dx + \mu \sum_{i=1}^N \int_{\Omega} |K(\phi_i)|^2 dx. \quad (18)$$

Here $\mu > 0$ is a penalisation parameter which usually is a fixed parameter chosen *a priori*, or it can in some cases be increased carefully through the iterations to improve the convergence. $\boldsymbol{\lambda} = \{\lambda_i\}_{i=1}^N$ is the Lagrangian multipliers, where λ_i is a function defined in the same domain as ϕ_i .

We search a saddle point of L and therefore require

$$\begin{aligned} \frac{\partial L}{\partial \phi_i} &= \frac{\partial \tilde{F}}{\partial \phi_i} + \lambda_i K(\phi_i) + 2\mu K'(\phi_i) = 0, \quad \forall i \in \{1, \dots, N\}, \\ \frac{\partial L}{\partial \lambda_i} &= K(\phi_i) = 0, \quad \forall i \in \{1, \dots, N\}, \\ \frac{\partial L}{\partial c_j} &= \frac{\partial \tilde{F}}{\partial c_j} = 0, \quad \forall j \in \{1, \dots, 2^N\}, \end{aligned} \quad (19)$$

where $\frac{\partial \tilde{F}}{\partial \phi_i}$ and $\frac{\partial \tilde{F}}{\partial c_j}$ are given in (16) and (17).

Starting with initial guesses ϕ^0 , $\mathbf{c}^0$ and $\boldsymbol{\lambda}^0$, we iterate towards the better approximations denoted by ϕ^k , $\mathbf{c}^k$ and $\boldsymbol{\lambda}^k$ where $k = \{1, 2, \dots\}$. These variables are updated using a steepest descent method, and when the change of the variables approach zero, the iterations can be stopped. In [30] a MBO operator splitting scheme has been applied for solving related problems. The actual applied algorithm is as follows:

Algorithm 1 (*Uzawas Algorithm for Variational Level Set Methods*)

Determine how many level set functions, N , to use.

Choose timestep for ϕ : Δt_ϕ .

Choose search interval for each c_j : $c_j \in [a_j, b_j]$.

Initialise: ϕ^0 , $\mathbf{c}^0$ and $\boldsymbol{\lambda}^0$ and set $k = 0$.

1. Update ϕ ;
 - a) Compute q by (13).
 - b) Evolve the level set functions: $\phi^{k+1} = \phi^k - \Delta t_\phi \frac{\partial L}{\partial \phi}(\phi^k, \mathbf{c}^k, \boldsymbol{\lambda}^k)$.
2. Update $\mathbf{c}$ (after a fixed number of iterations);

For each c_j , $j = 1, 2, \dots, 2^N$:

 - a) Compute q by (13).
 - b) Define: $\alpha_{c_j}^k = \frac{\partial L}{\partial c_j}(\phi^{k+1}, \mathbf{c}^k, \boldsymbol{\lambda}^k)$.
 - c) Define the search interval: Let $M \in \mathbb{R}$ be all values of Δt such that $c_j^k - \Delta t \alpha_{c_j}^k \in [a_j, b_j]$.
 - d) Find the optimal timestep: $\Delta t_{c_j} = \arg \min_{\Delta t \in M} L(\phi^{k+1}, \mathbf{c}^k - \Delta t \alpha_{c_j}^k \mathbf{e}_j, \boldsymbol{\lambda}^k)$, where $\mathbf{e}_j$ is the j 'th unit vector.
 - e) Update this constant: $c_j^{k+1} = c_j^k - \Delta t_{c_j} \alpha_{c_j}^k$.
3. Update $\boldsymbol{\lambda}$ (after a fixed number of iterations);

$$\boldsymbol{\lambda}^k = \boldsymbol{\lambda}^k + \mu \mathbf{K}(\phi^{k+1}).$$
4. Iterate again if necessary;

$$k = k + 1.$$

Notice that q is updated implicitly using the most recently calculated values of ϕ and $\mathbf{c}$. In this algorithm we do not use step 2 and 3 in every iteration. This is because the algorithm becomes unstable if $\mathbf{c}$ and $\boldsymbol{\lambda}$ are updated too often. In principle we could have run step 1 to convergence before doing the other steps. Numerically this is not strictly necessary and it would have been computationally heavy. We have therefore updated $\mathbf{c}$ and $\boldsymbol{\lambda}$ after a fixed numbers of iterations.

6 Numerical Results

In this section we will present some numerical examples where we study the performance of the presented method. The studied examples are synthetic cases where the true permeability field consists of two distinct permeability values, and in these cases it is sufficient with one level set function to represent the field. Cases with more than two distinct permeability values require more level set functions. In that case the solution approach will be

Table 1. Properties for the simulations.

Reservoir dimensions:	1000 × 1000 × 40 meter	
Simulation grid:	16 × 16 × 1 cells	
Porosity:	0.2	
Viscosity:	$\mu_w = 0.5 \cdot 10^{-3}$ Pa s	$\mu_o = 0.5 \cdot 10^{-3}$ Pa s
Endpoint relative permeabilities:	$\hat{\kappa}_{rw} = 0.1$	$\hat{\kappa}_{ro} = 1$
Residual saturations:	$S_{rw} = 0.2$	$S_{or}^* = 0.2$
Corey exponents:	$e_w = 1.5$	$e_o = 2.5$
Initial saturation:	$S_w = 0.2$	$S_o = 0.8$
Capillary pressure function:	$P_c(S_w) \equiv 0$ kPa	
Injection rate (Ex. 1):	8% of total pore volume per year.	
Injection rate (Ex. 2 and 3):	3.5% of total pore volume per year.	
Production rate:	constant BHP = 200.0 bar	
Number of timesteps:	192	
Total production time:	3000 days	
Number of seismic surveys:	16 (i.e., approximately every 6 months.)	

more time-consuming, and the larger number of parameters will add additional difficulties in the optimisation process. In [24, 8] multiple level set representations are tested on elliptic inverse problems. The forward model in the history-matching problem is computationally much more demanding than for the elliptic problem. Due to the large time-consumption and the additional difficulties, multiple level set representations are not tested numerically in this paper, but should be an issue for further work.

The test reservoir is square and horizontal with constant thickness and no-flow outer boundaries. Except for the absolute permeability, the fluid and rock properties are held fixed through the simulations. In the field we have one injector positioned in the lower left corner, and one producer positioned in the upper right corner.

The relative permeability functions are defined by the Corey models;

$$\begin{aligned}\kappa_{rw} &= \hat{\kappa}_{rw} \left(\frac{S_w - S_{wr}}{1 - S_{wr} - S_{or}} \right)^{e_w}, \\ \kappa_{ro} &= \hat{\kappa}_{ro} \left(\frac{S_o - S_{or}}{1 - S_{or} - S_{wr}} \right)^{e_o},\end{aligned}$$

where the Corey exponents, e_w and e_o , the residual saturations, S_{wr} and S_{or} , and the endpoint permeabilities, $\hat{\kappa}_{rw}$ and $\hat{\kappa}_{ow}$, are assumed known. The numerical values for these properties are, together with the rest of the properties for the simulations, listed in Table 1.

The forward model (the solution of (1)–(4) for a given function $q(\mathbf{x})$) is solved by applying an in-house reservoir simulator. In the simulator the equation error is minimised by applying Newton iterations, and the linear solver of choice is GMRES. The gradients, $\frac{\partial F}{\partial q}$, are obtained from the solution of the adjoint system of equations, see e.g., [17, 23].

Table 2. Standard deviations for the added noise. The noise is larger for the seismic data than for the well data.

	Well data	Seismic data
Pressure	$\sigma_{p,\text{well}} = 1.0$ bar	$\sigma_{p,\text{seis}} = 2.5$ bar
Saturation	$\sigma_{S_w,\text{well}} = 0.025$	$\sigma_{S_w,\text{seis}} = 0.050$

For each reference permeability field we calculate the true values of saturation (S_w) and pressure (p_o) for the applied timesteps on the given grid. Thereafter synthetic measurements are constructed by adding noise to the calculated true values. The noise is assumed to be uncorrelated Gaussian noise with zero mean. In Table 2 the standard deviations which give the amount of added noise are listed. Notice that the uncertainties are larger for the seismic measurements than for the measurements in the wells. The noise is generated with the same covariance matrices as used in the objective function. This gives a natural weight between the terms J_{well} and J_{seis} according to the probabilistic approach described in Section 2. In a real case, an estimate of the respective noise levels could be applied when constructing the covariance matrices, however, uncertainties in these estimates may force a non-optimal weight between different measurements.

The amount of seismic data are in the presented examples quite rich. Study of how less seismic data will influence the final result can be found in [23].

In all numerical examples in this paper we have one injector (where water is injected) in the lower left corner, and one producer in the upper right corner. Between the two wells there is a high permeability channel with possibly different shapes for the different examples. The setup with the wells and the high permeable channel connecting the two wells are not critical for the method to work. In other papers we have investigated cases with more wells [23] and with other structures of the permeability field [25, 23]. The performance of the method, and the possible complexity of the solution, is however dependent of the content of information from the data. If more wells are present, more information is generally available from the well data, and a more detailed description of the solution may be possible to obtain. The content of information from the data may also be dependent of the water flooding in the domain.

The penalisation parameter μ is increased slowly through the iterations. If k is the number of iterations, $\mu = 0.05 \cdot 1.01^k$ up till it reaches an upper bound (equal to 4) where we keep it fixed. Regarding the regularisation parameter β , we are for each example first trying with a value of $5 \cdot 10^{-3}$. If this causes large oscillations in the solution, then the weight on the regularisation is increased and a new optimisation is preformed. Both the Lagrangian parameter λ and the c_j -values are updated each 10^{th} iteration.

For each test case we start with $\phi^0 = 0$ in the entire domain except in the cells where we have wells. If no prior information is added, an initial $\phi^0 = 0$ means that we do not assume anything about the contours of the

discontinuities. In the cells with a penetrating well, we assume that the approximate permeability value is known. The value of ϕ is therefore fixed equal to its correct value (-1 or 1 dependent of the initial $\mathbf{c}$ -value) in these cells.

For each of the constant values we define an interval $[a_j, b_j]$ within c_j should be estimated. The length of this interval will correspond to the prior uncertainty of the permeability value for the corresponding region. Because there are abilities for direct measurements of the permeability in the wells, we have applied a lower uncertainty for c_j in the regions where there is at least one well present, than for the regions with no wells. For the studied cases we have applied intervals $[a_j, b_j]$ with length equal 50% (no wells) and 30% (wells) of the difference between the two true values of q . The centre of the intervals are chosen equal to the true values. For example, if we assume the following; c_1 and c_2 are the true values, the region corresponding to c_1 has no wells present and there are one or more wells penetrating a region with permeability approximately equal to c_2 . Then the bounds will be

$$a_1 = c_1 - 0.25 \cdot |c_2 - c_1|, \quad b_1 = c_1 + 0.25 \cdot |c_2 - c_1|$$

and

$$a_2 = c_2 - 0.15 \cdot |c_2 - c_1|, \quad b_2 = c_2 + 0.15 \cdot |c_2 - c_1|.$$

In this work we start with initial c_j -values on the lower and upper bound of the two intervals. We use the lower bound for the smallest c_j -value and the upper bound for the highest c_j -value, that is, if $c_1 < c_2$, then $c_1^0 = a_1$ and $c_2^0 = b_2$. Other approaches for choosing the initial values are also possible.

In the last example presented in this paper we have incorporated prior information in the parameter estimation process. J_{prior} will then be non-zero and the matrix D_{prior} will give the weight of the prior term in the objective function. In this example we have assumed that the spatial correlation length in the prior model q_{prior} is smaller than the size of the grid cells, i.e., D_{prior} is diagonal with entries equal to σ_{prior}^2 , where σ_{prior} is the standard deviation of the values in $q_{\text{prior}}(\mathbf{x})$. A non-diagonal matrix may be included as shown in e.g., [17]. In this work we have used

$$\sigma_{\text{prior}} = \frac{1}{2} \max_{i \neq j} |a_i - b_j|, \quad i, j \in \{1, 2\},$$

where a_i and b_j are the bounds defined above. The magnitude of σ_{prior} will now correspond to half the distance between the upper and lower bound of the estimated function q^k .

The algorithm is stopped after 1000 iterations if ϕ^k and $\mathbf{c}^k$ have not converged (in the sense of stopped changing) before this.

To measure the data fit we plot RMS values of $J_{\text{well+seis}}$ and J_{prior} versus the iteration number. Here $J_{\text{well+seis}} = J_{\text{well}} + J_{\text{seis}}$. The RMS value of a function J_γ is defined as $\sqrt{2J_\gamma/n_\gamma}$, where n_γ is the number of measurements included in J_γ .

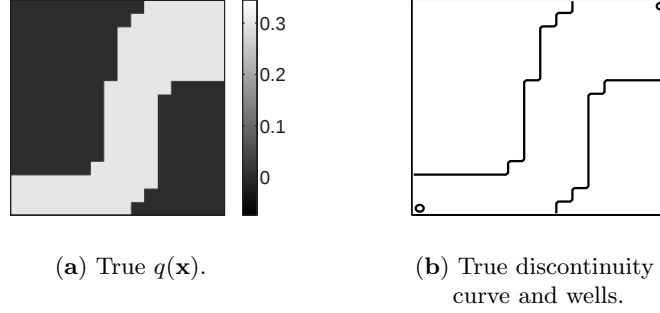


Fig. 2. Example 1: True permeability and the corresponding discontinuity. The constant levels are given by $\mathbf{c} = (0, 0.3)$, which corresponds to a permeability equal to 1 D and 2 D. The circles in the corners are indicating the positioning of the wells.

Another measure applied to check the convergence is $\|K(\phi^k)\|_{L_2(\Omega)}$, which indicates how fast ϕ^k reaches the convergence values -1 and 1.

In the first presented example, see Section 6.1, we will illustrate how the method is working by showing an optimisation of a field which the model is able to reproduce with a relatively low error without incorporating prior information. In Example 2 (Section 6.2) and 3 (Section 6.3) we will show estimations of a field which is more difficult to reproduce. The field and the setup for these two examples are equal except that prior information is added in Example 3 and not in Example 2. The comparison of these estimations will show how prior information can influence the estimate.

6.1 Example 1: S-shaped Channel

The true field for this example is an S-shaped channel with high permeability from the injector to the producer. A plot of the field is shown in Figure 2(a), while the true discontinuity curve of the permeability is plotted in Figure 2(b). In the field there are three distinct piecewise constant regions, but since two of the regions have the same constant value, *one* level set function is sufficient to give a representation of it. This is related to the level set methods' nice feature of splitting and merging regions independent of their contours (see for example [29]). In this example no prior information is added in the optimisation process.

In Figure 3 the development of the estimates q^k and the signchange of ϕ^k are shown. Already after 50 iterations the estimate is quite close to the true field, but we need approximately 200 iterations to produce a field which is piecewise constant with only two levels. From this time the solution stops changing.

Error measures and convergence curves are shown in Figure 4. If we compare the different curves, we observe that the RMS function of $J_{well+seis}$ (Figure 4(c)) is decreasing much faster and for a shorter period than what is

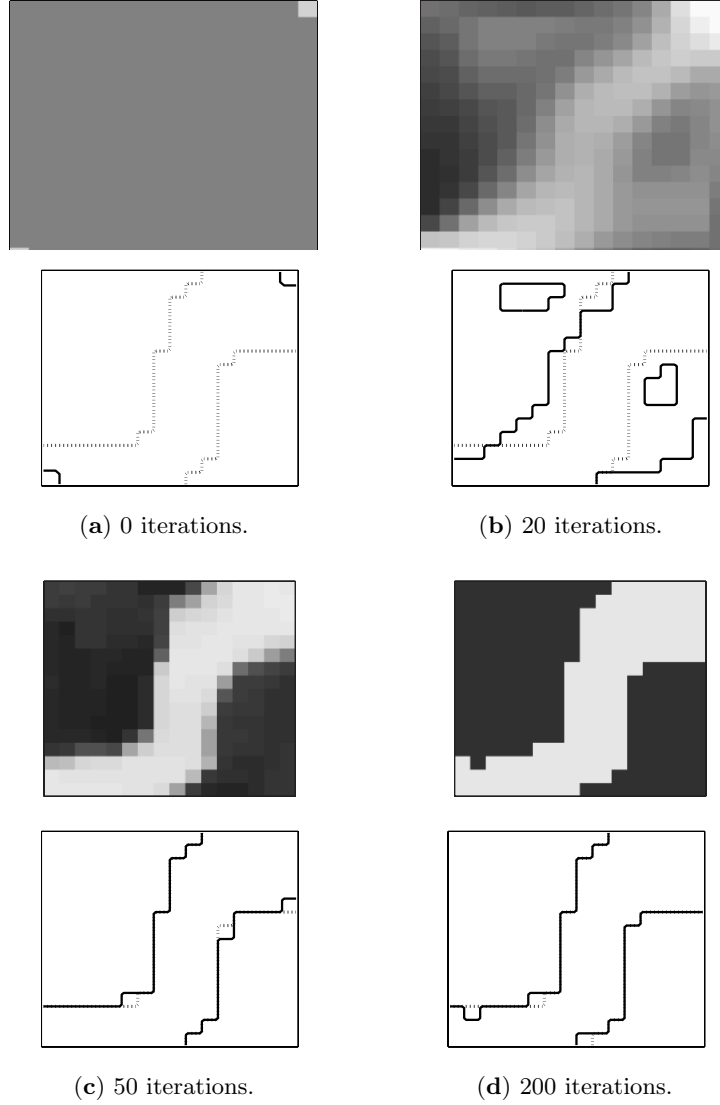


Fig. 3. Example 1: The estimated permeability for different iterations. In the upper row q^k is plotted with the same colourmap as used in Fig. 2(a). In the lower rows the signchanges of ϕ^k are shown by the solid lines, and the discontinuities of the true $q(\mathbf{x})$ are given by the dotted lines. Initially $\phi^0 = 0$ in the entire domain, except in the corners where the wells are located. In the intermediate iterations the values of ϕ^k will evolve towards -1 or 1 in the different parts of the domain. After about 50 iterations the true field is approximately matched. We though need about 200 iterations before ϕ^k is (approximately) equal to 1 or -1 in all cells, and at this stage the field is piecewise constant with only two levels.

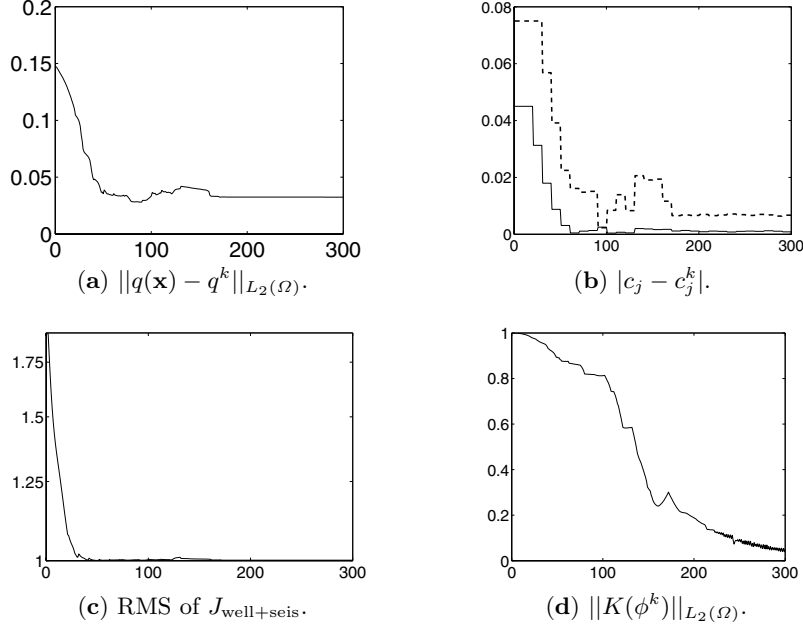


Fig. 4. Example 1. Error measures and convergence plots versus the iteration number. Figure (a) and (b) give the error in the computed q^k and c_j^k -values, respectively, and in Figure (c) the RMS values of $J_{\text{tot}} = J_{\text{seis+well}}$ are plotted. A measures of the convergence of ϕ^k is shown in Figure (d). The curves indicate convergence after about 200 iterations.

the case for all the other functions. After the initial rapid decrease, the RMS function reaches a stable value just above 1. The other measures are also reaching stable values, but after a higher number of iterations. Notice that the RMS function is plotted in semilogarithmic scale, while the other functions are given in linear plots. This makes the difference in the behaviour of the curves even more clear.

The rapid decrease in the RMS function of $J_{\text{well+seis}}$ can usually be explained by low sensitivities with respect to the permeability changes in some areas of the field. The sought solution may therefore be difficult to find, and the convergence can be very slow towards the end of the optimisation. The described phenomenon illustrates the ill-posedness of the treated inverse problem.

6.2 Example 2: System of Channels

This example involves a more complicated field where two channels are crossing each other (see Figure 5). The two channels are assumed to have the same permeability value, and together they produce a connected region with high

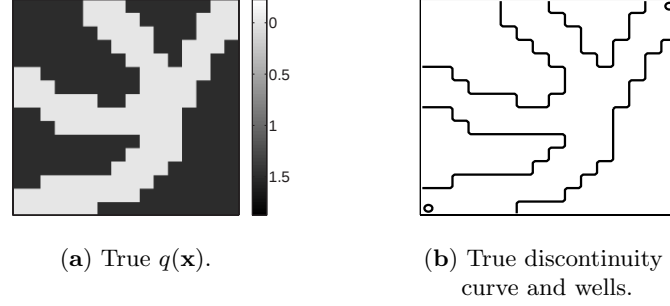


Fig. 5. Example 2: True permeability and the corresponding discontinuity. The constant levels are given by $\mathbf{c} = (-1.5, 0)$, which corresponds to a permeability equal to 32 mD and 1 D. The circles in the corners are indicating the positioning of the wells.

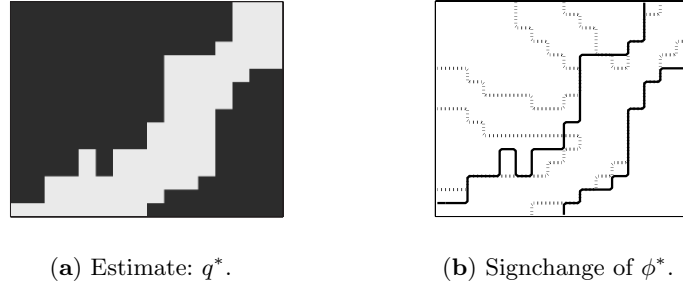


Fig. 6. Example 2: The estimated permeability. In Figure (a) q^* is plotted with the same colourmap as used in Figure 5(a). The algorithm is ran 1000 iterations. In Figure (b) the signchanges of ϕ^* are shown by the solid lines, and the discontinuities of the true $q(\mathbf{x})$ are given by the dotted lines. Only the parts of the channels in the main flow direction is recovered.

permeability from the injector to the producer. No prior information is added in the optimisation process.

The final result in Figure 6 shows that we are capturing a high permeable channel from the injector to the producer. The other branches of the channels of the true field are not discovered. In some of the convergence plots in Figure 7 we can observe large oscillations. This illustrates the difficulties of producing a piecewise constant field as a solution to this problem. The relative reduction in the error of q^k (Figure 7(a)) is quite small for this example. This can be explained by the misclassification of some parts of the channels. The misclassified parts are by the method classified as low permeable regions, which in fact is less close to the true solution than the initial guess. The initial value, q^0 , is (except for the cells with wells) equal to the mean of c_1^0 and c_2^0 (given by $\phi = 0$ in (11)).

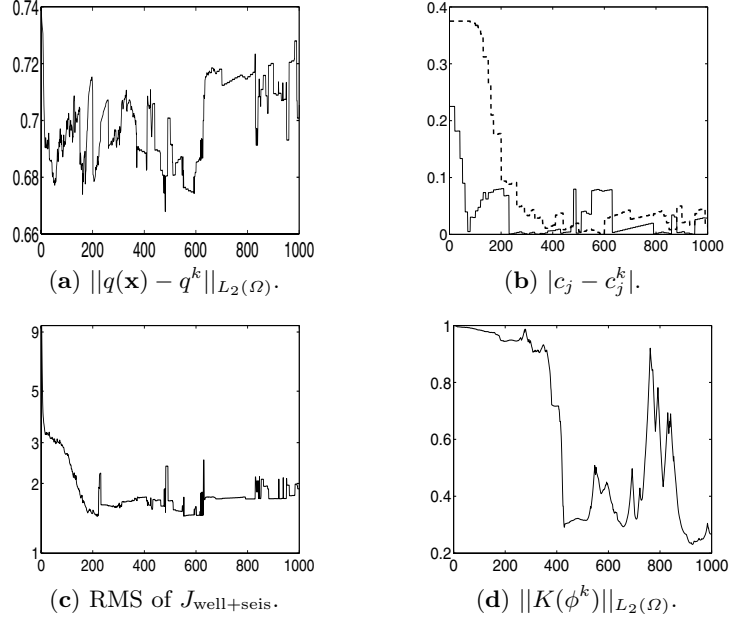


Fig. 7. Example 2. Error measures and convergence plots versus the iteration number. Figure (a) and (b) give the error in the computed q^k and c_j^k -values, respectively, and in Figure (c) the RMS values of $J_{\text{tot}} = J_{\text{seis}+\text{well}}$ are plotted. A measure of the convergence of ϕ^k is shown in Figure (d).

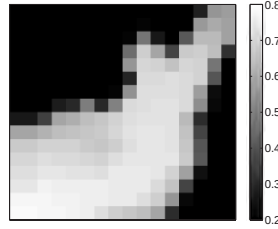


Fig. 8. The saturations, S_w , of the true field of Example 2 and 3 at the end of the simulation (after 3000 days).

In Figure 8 we have plotted the simulated values of S_w for the true field at the end of the simulation. This plot shows that only parts of the field is flooded by water at this time level. The main part of the flow will go in the high permeable region discovered by the level set method. That is, the flow will move very slowly in the low permeable regions and also in the parts of the channels which are not discovered by the optimisation process. In regions where there is almost no change in S_w , the amount of information from these

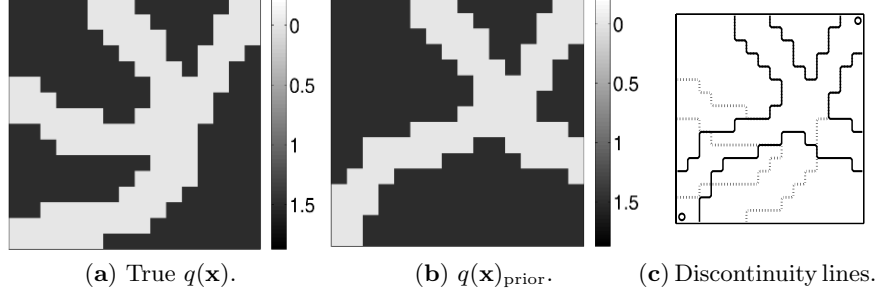


Fig. 9. Example 3: True permeability and the prior model. In Figure (c) the discontinuity lines for the true $q(\mathbf{x})$ are shown by the dotted lines and the discontinuities of $q(\mathbf{x})_{\text{prior}}$ are shown by solid lines. The circles in the corners are indicating the positioning of the wells. The constant values for both the true $q(\mathbf{x})$ and q_{prior} are given by $\mathbf{c} = (-1.5, 0)$, which corresponds to a permeability equal to 32 mD and 1 D.

data will be low, and thus, the sensitivities with respect to changes in the permeability may also be low. According to this analysis, it is reasonable that the structures of the field are more difficult to reproduce in regions where the change in S_w is low, than in other parts of the reservoir. A more thorough discussion about the information from the seismic data and more examples illustrating this issue can be found in [25].

6.3 Example 3: System of Channels - with Added Prior Information

In this example we will recover exactly the same field as in Example 2, but in this case we will add prior information in the optimisation process. The field used as prior information for $q(\mathbf{x})$ is given in Figure 9(b). In Figure 9(c) a comparison of the discontinuity lines of the true $q(\mathbf{x})$ and the prior model $q_{\text{prior}}(\mathbf{x})$ is shown. Compared to the true field, the prior model is matching parts of the high permeable channels, but other parts are clearly misclassified. The values of the constants in $q_{\text{prior}}(\mathbf{x})$ are for simplicity chosen equal to the constants in the true field.

From the recovered field given in Figure 10 (d), we observe, as in Example 2, that the main structures of the channel from the injector to the producer is recovered quite easily. Notice that the part of this channel closest to the injector is different from the channel in the initial model, see Figure 9 (c). In this part of the reservoir the change in S_w through the production history for the true $q(\mathbf{x})$ is high (Figure 8), and we therefore probably have larger amount of useful information in this region than in other parts of the reservoir. In the parts of the reservoir with a low change in S_w (for the true $q(\mathbf{x})$) we observe that the solution is keeping the structures from the prior model. This means that if a proper weight is chosen for the prior term in the objective function,

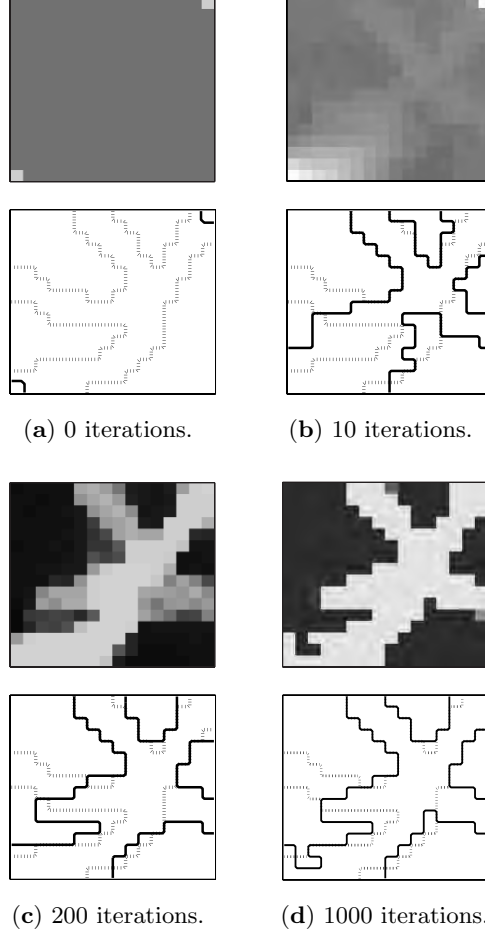


Fig. 10. Example 3: The estimated permeability for different iterations. In the upper figures q^k is plotted with the same colourmap as used in Figure 9(a) and (b). In the lower figures the signchanges of ϕ^k are shown by the solid lines, and the discontinuities of true $q(\mathbf{x})$ are given by the dotted lines.

the prior model is filling out the information from the measured data in the regions where this information is deficient in order to capture the permeability trends. In the parts of the field where the information from the data clearly is sufficient to reproduce the structures of the true field, the estimate will, in order to reconcile the data, not necessarily follow the prior model.

The error and convergence plots for this example are given in Figure 11. Comparing these plots to the corresponding plots from Example 2 (Figure 7) we observe a lower error for the computed function q^* when the prior information is added. In this case, this is due to a better match of the geometries

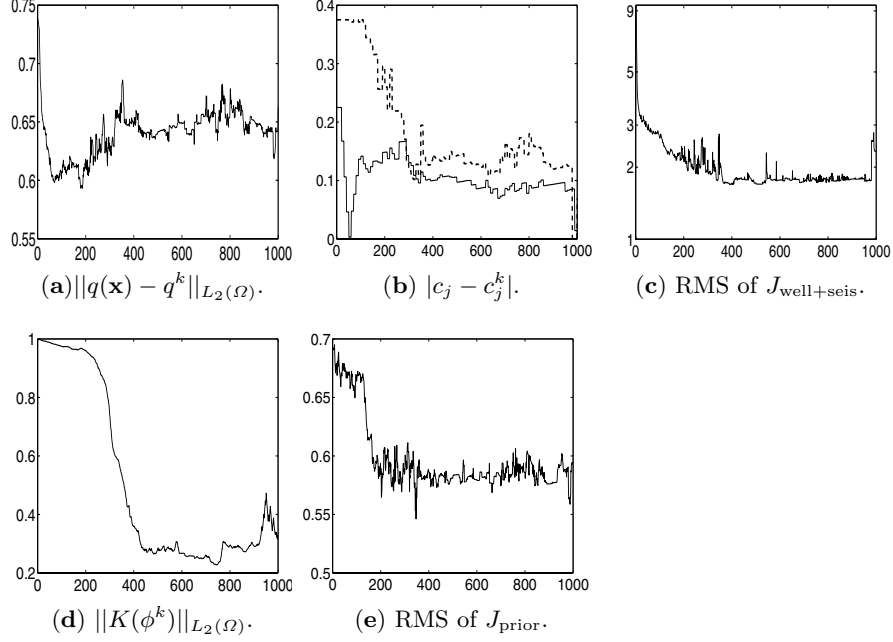


Fig. 11. Example 3: Error measures and convergence plots versus the iteration number. Figure (a) and (b) give the error in the computed q^k and c_j^k -values, respectively, and in Figure (c) the RMS values of $J_{\text{seis}+\text{well}}$ are plotted. A measure of the convergence of ϕ^k is shown in Figure (d). In Figure (f) the RMS values of J_{prior} are plotted.

of the channels. In spite of the better match of the geometries, the errors of the constant values c_j^* are larger in the last example.

In Figure 11(e) the RMS function of J_{prior} is shown. The curve decreases in the beginning, before it starts to oscillate around a stable value. Since this is a regularisation term, this measure is not intended to approach zero unless the true field is very close to the prior model, or the information from the data is insufficient to reproduce structures not recovered in the prior model.

7 Summary and Conclusions

We have applied a binary level set formulation for solving inverse two phase porous media flow problems. Both well data and seismic time-lapse data are utilised in the optimisation process. In addition, we can incorporate prior information about the sought solution. The estimated model's ability to reproduce the measured data and the closeness of the estimated model to the prior information are measured in one objective function.

The method is searching a piecewise constant solution of the inverse problem, and it is regularised by a total variation norm. The geometries of the discontinuity curves are allowed to have arbitrary shapes only controlled by the total variation regularisation.

The numerical studies focus on piecewise constant permeability fields with two different constant levels. The presented examples show that the method is able to recover the main structures of permeability fields even with rather complicated systems of channels. This conclusion is also supported by the results from [25]. Misclassifications of regions seem to be due to lack of information from the measured data in certain parts of the domain.

If prior information is added in the optimisation process (and weighted properly), the estimate will to a large extent follow the prior model in regions where the amount of information from the measured data is low, and at the same time be able to change the field in a correct direction in other parts of the reservoir. In the parts with large amount of information, the method will, if necessary, change the structures of the estimate away from the prior model such that the fit to the measured data will be improved.

8 Acknowledgements

We gratefully acknowledge Daniel Christopher Doublet and Raymond Martinsen for providing their code for the forward reservoir simulator including gradient calculations, and for their help related to running this.

9 Nomenclature

S	= saturation
p	= pressure
Φ	= porosity
l	= fluid phase
μ_l	= viscosity for fluid l
κ	= permeability
κ_r	= relative permeability
f	= external volumetric flow
t	= time
$\mathbf{x}$	= spatial position
Ω	= reservoir domain
$\hat{\kappa}_r$	= endpoint permeability
e	= Corey exponent
S_r	= residual saturation
q	= logarithmic permeability
J	= objective function
$\mathbf{d}$	= vector of observed data

$\mathbf{m}$ = vector of output from simulator
 ϕ = level set functions
 $\mathbf{c}$ = constant levels of q
 F = regularised functional to be minimised
 L = Lagrangian functional
 λ = Lagrangian multiplier
 μ = penalisation constant
 β = regularisation parameter
 $\mathbf{K}$ = constraint for the optimisation
 σ = standard deviation
 k = iterations
 a_j = lower bound of c_j
 b_j = upper bound of c_j
 α = search interval

9.1 Subscripts

w = water
 o = oil
well = contribution from wells
seis = contribution from seismic
prior = contribution from prior information
tot = total contribution
 j = index of elements in $\mathbf{c}$
 i = index of level set functions

9.2 Superscripts

T = transpose
 k = iterations
 N = number of level set functions

References

1. S.I. Aanonsen, I Aavatsmark, T Barkve, A Cominelli, R Gonard, O Gosselin, M Kolasinski, and H Reme. Effect of scale dependent data correlations in an integrated history matching loop combining production data and 4d seismic data. In *Proceedings of the SPE Reservoir Simulation Symposium*, Houston, Texas, Feb. 2003. SPE 79665, presented at the SPE Reservoir Simulation Symposium held in Houston, Texas.
2. S.I. Aanonsen, A Cominelli, O Gosselin, I Aavatsmark, and T Barkve. Integration of 4D Data in the History Match Loop by Investigating Scale dependent Correlations in the Acoustic Impedance Cube. In *Proceedings of the 8th European Conference on the Mathematics of Oil Recovery*, Freiberg, Germany, 3-6 Sept. 2002.

3. U. Ascher and E. Haber. Grid refinement and scaling for distributed parameter estimation problems. *Inverse Problems*, 17:571–590, 2001.
4. U. Ascher and E. Haber. Computational methods for large distributed parameter estimation problems with possible discontinuities. *Symp. Inverse Problems, Design and Optimization*, 2004.
5. U. Ascher, E. Haber, and H. Huang. On effective methods for implicit piecewise smooth surface recovery. Submitted 2004.
6. M. Burger. A level set method for inverse problems. *Inverse problems*, 17:1327–1355, 2001.
7. M. Burger and S. Osher. A survey on level set methods for inverse problems and optimal design. UCLA, CAM-Report 04-02, 2004.
8. T. Chan and X.-C. Tai. Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients. *Journal of Computational Physics*, 193:40–66, 2003.
9. T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. Image Processing*, 10(2):266–277, 2001.
10. E. Chung, Chan T., and X.-C. Tai. Electrical impedance tomography using level set representation and total variational regularization. submitted, 2004.
11. O. Dorn, E. Miller, and C. Rappaport. A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets. *Inverse Problems*, 16:1119–1156, 2000. Special issue on Electromagnetic Imaging and Inversion of the Earth's Subsurface.
12. G. Chavent and J. Liu. Multiscale parameterization for the estimation of a diffusion coefficient in elliptic and parabolic problems. In *Proceedings of the Fifth IFAC Symposium on Control of Distributed Parameter Systems*, Perpignan, France, June 1987.
13. F. Gibou and R. Fedkiw. Fast hybrid k-means level set algorithm for segmentation. *Stanford Technical Report*, November 2002.
14. O. Gosselin, S.I. Aanonsen, I. Aavatsmark, A. Cominelli, R. Gonard, M. Kolasinski, F. Ferdinandi, and K. Kovacic, L. and Neylon. History matching Using Time-lapse Seismic (HUTS). In *Proceedings of the SPE Annual Technical Conference and Exhibition*, Denver, Colorado, 30 Sept.-3 Oct. 2003. SPE 84464.
15. K. Ito, K. Kunisch, and Z. Li. Level-set function approach to an inverse interface problem. *Inverse problems*, 17:1225–1242, 2001.
16. M. Landrø and Ø Kvam. Pore Pressure Estimation - what can we learn from 4D. *CSEG Recorder*, September 2002.
17. R. Li, A.C. Reynolds, and D.S. Oliver. History matching of three-phase flow production data. *SPE Journal*, 8(4), December 2003.
18. J. Lie, M. Lysaker, and X.-C. Tai. A piecewise constant level set framework. url:“<http://www.mi.uib.no/BBG/papers.html>”, 2004.
19. J. Lie, M. Lysaker, and X.-C. Tai. A piecewise constant level set level set framework. In *Proceedings of European Congress on Computational Methods in Applied Sciences and Engineering*, Jyväskylä, July 2004.
20. J. Lie, M. Lysaker, and X.-C. Tai. A binary level set model and some applications to mumford-shah image segmentation. Accepted and to appear in *IEEE Transaction on image processing*, 2005.
21. D.G. Luenberger. *Optimization by Vector Space Methods*. Wiley Professional Paperback Series. Wiley, 1969.

22. M. Lygren, K. Fagervik, T.S. Valen, A. Hetlelid, G. Berge, G.V. Dahl, L. Snneland, H.E. Lie, and I. Magnus. A method for performing history matching of reservoir flow models using 4d seismic data. *Petroleum Geoscience*, 9:85–90, 2003.
23. L.K. Nielsen. *Reservoir Characterisation by a Binary Level Set Method and Adaptive Multiscale Estimation*. PhD thesis, Department of Mathematics, University of Bergen, 2006.
24. L.K. Nielsen, X.-C. Tai, S. Aanonsen, and M. Espedal. A binary level set model for elliptic inverse problems with discontinuous coefficients. UCLA, CAM-Report 05-51, 2005.
25. L.K. Nielsen, X.-C. Tai, S.I. Aanonsen, and M. Espedal. Reservoir description using a binary level set model. UCLA, CAM-Report 05-50, 2005.
26. S. Osher and J.A. Sethian. Fronts propargating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comput. Phys*, 79(1):12–49, 1988.
27. F. Santosa. A level-set approach for inverse problems involving obstacles. *ESAIM: Contr. Optim. Calc. Var.*, 1:17–33, 1996.
28. B. Song and T. Chan. A fast algorithm for level set based optimization. *UCLA CAM-Report 02-68*, 2002.
29. X.-C. Tai and T. Chan. A survey on multiple level set methods with applications for identifying piecewise constant functions. *Int. J. of numerical analysis and modeling*, 1(1):25–47, 2004.
30. X.-C. Tai, O. Christiansen, P. Lin, and I. Skjaelaaen. A remark on the mbo scheme and some piecewise constant level set methods. UCLA, CAM-Report 05-24, 2005.
31. O. Yilmaz and S.M. Doherty, editors. *Seismic data analysis: processing, inversion, and interpretation of seismic data*. Society of Exploration Geophysicists, Tulsa, 2001.

A

Color Figures

From “Image Inpainting Using a TV-Stokes Equation,”
by Tai, Osher, and Holm



Fig. A.1. The original image.

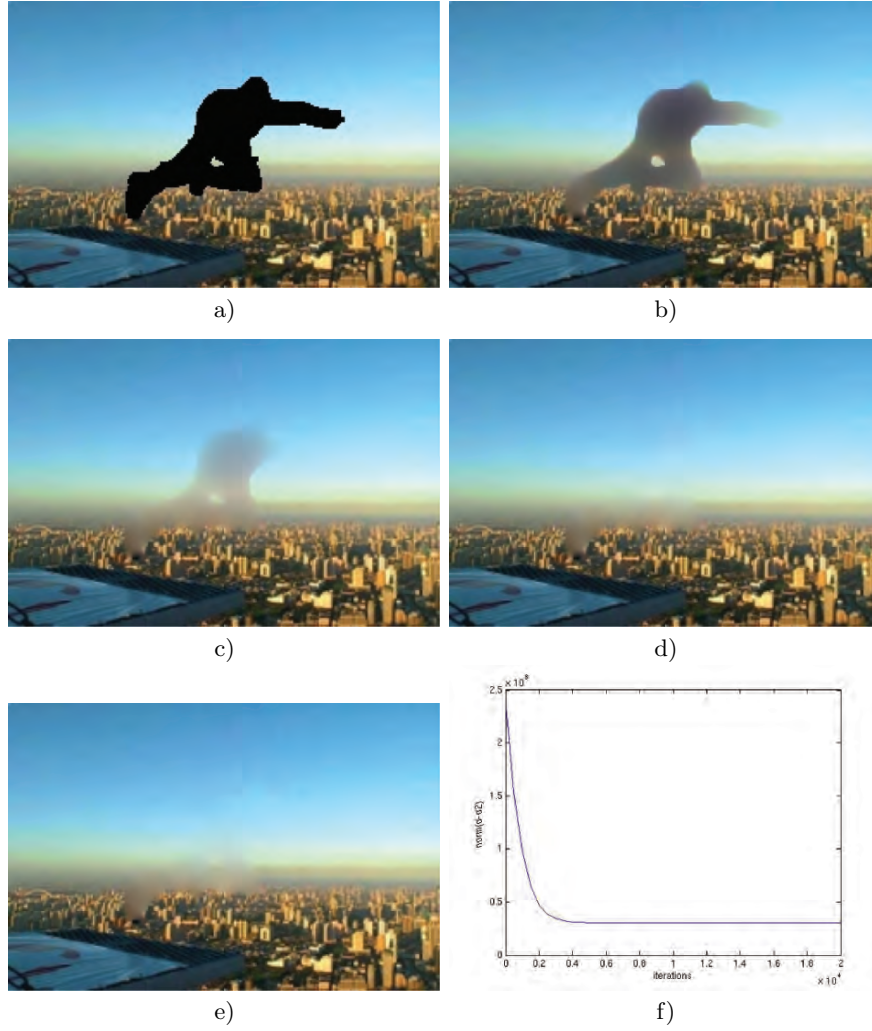


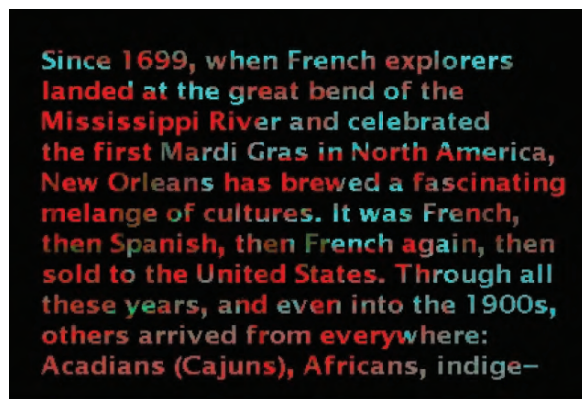
Fig. A.2. The restored image d using equation (10) at different iterations. a) at iteration 0; b) at iteration 10000; c) at iteration 20000; d) at iteration 30000; e) The restored image using the new method (15)-(16) to find τ . f) The plot for $\|d - d_0\|$ which shows that the equation (5) reaches a steady state, i.e., at iteration 30000. f) The plot for $\|\tau^{n+1} - \tau^n\|$ which goes to zero very quickly which also shows the steady state is quickly reached.



a)



b)



c)

Fig. A.3. a) The original image. b) The restored image using equations (5) and (10). c) The difference image.

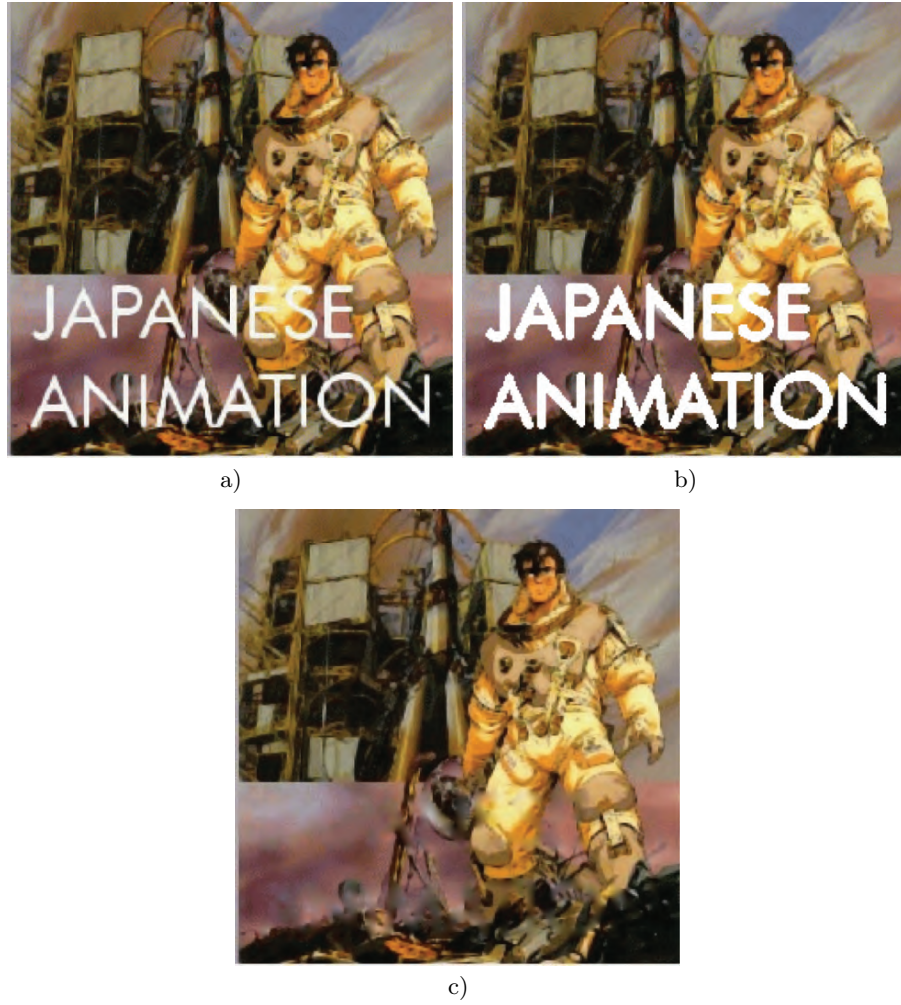


Fig. A.4. a) The original image. b) The image with the inpainting region obtained manually. c) The restored image using equations (5) and (10).



a)



b)

Fig. A.5. a) The image with the inpainting region white. b) The restored image using equations (5) and (10).



Fig. A.6. An image from the match between Norway and Croatia in the XIX Men's World Championship.



a)



b)



c)

Fig. A.7. a) The restored image using Dirichlet boundary conditions. b) The image with the inpainting region violet. c) The restored image using Dirichlet and Neumann boundary conditions.

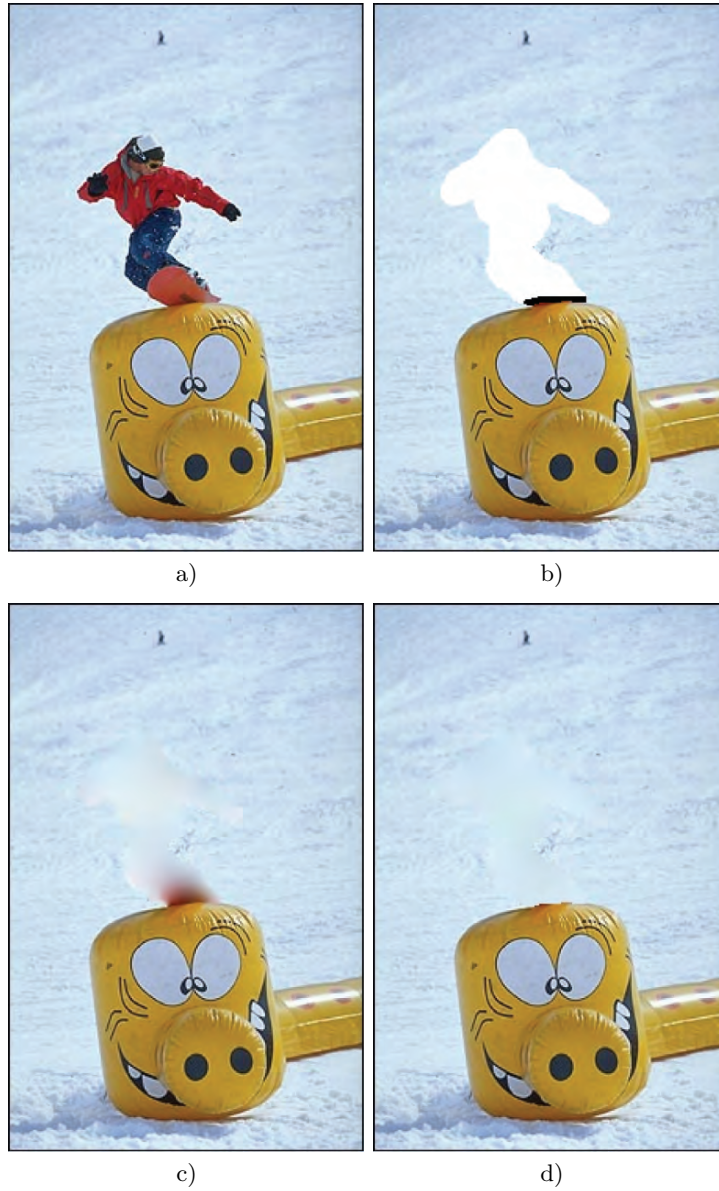


Fig. A.8. a) A photo taken by Espen Lystad, a well-known snowboard photographer in Norway. b) The image with the inpainting region marked. The Neumann boundary is black. c) The restored image only using Dirichlet boundary condition. d) The restored image using Dirichlet and Neumann boundary conditions.

From “Image Dejittering Based on Slicing Moments,” by Kang and Shen



Fig. A.9. (a) Ideal image $u(x, y)$. (b) Randomly jittered image $u_J(x, y)$.

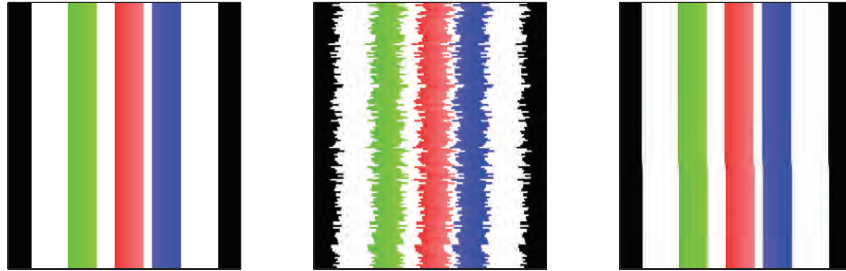


Fig. A.10. (a) Ideal image u . (b) Jittered image u_J . (c) Dejittered image u^* via moment regularization.

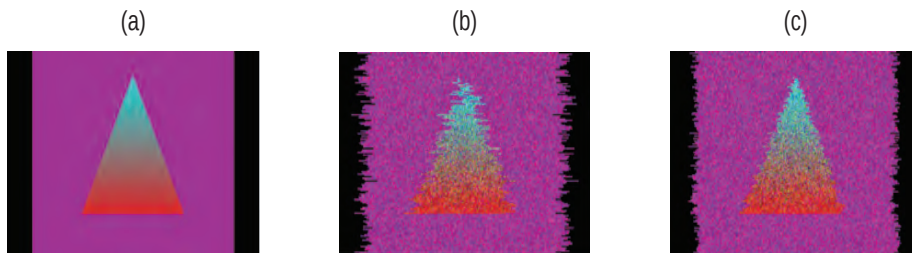


Fig. A.11. Ideal image u is with intensity Gaussian white noise three vertical bars. (a) Original image u , (b) Jittered image u_J . (c) Dejittered image. The de-jittered estimation in (c) shows the robustness of our model to the perturbation of intensity noises.

From “Chromaticity Denoising using Solution to the Skorokhod Problem,” by Borkowski



Fig. A.12. Chromaticity denoising. Top-bottom: original, noisy, denoised.

From “Some Recent Developments in Variational Image Segmentation,” by Chan, Moelich, and Sandberg

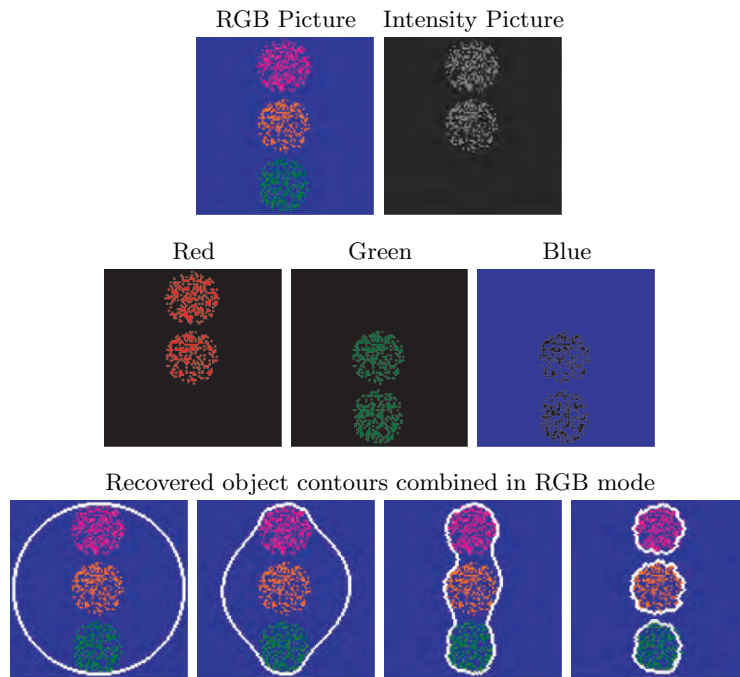


Fig. A.13. We give here an example of a color image that has three objects of different colors, while the corresponding gray scale image only shows two of them. The boundary of all the circles is found, while in the gray-scale image the boundary of one of the circles would never be detected. Note that, since this image does not have gradient edges, a gradient-based algorithm would not be able to find the three objects. The parameters are as follows: $\mu = 0.06 \cdot 255^2$, $\lambda_i^+ = \lambda_i^- = 1$, for $i = 1, 2, 3$.



Fig. A.14. Results of tracking an object using a modified version of the Chan–Vese algorithm.



Fig. A.15. Illustration of how algorithm handles position errors. The child moved far from frame to frame, by enlarging the contour the child is found in the following image.



Fig. A.16. Tracking in presence of background clutter and poor contrast.

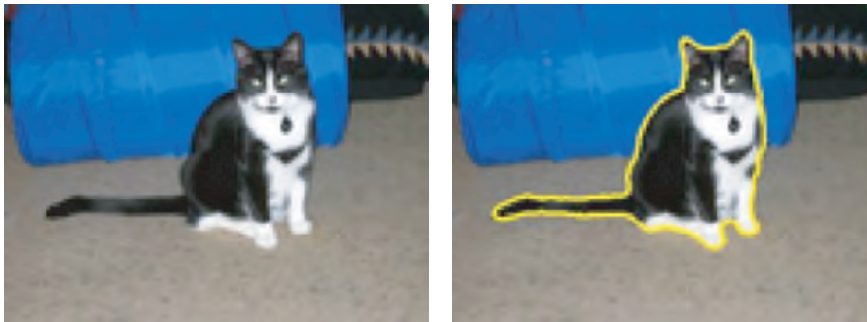


Fig. A.17. A black and white cat and output of color logic model.



Fig. A.18. Additional example of color logic model.



Fig. A.19. Illustration of improved background model. Choosing three colors (left) or two colors (right) with first background model, and choosing three object colors and three background colors for improved background model.

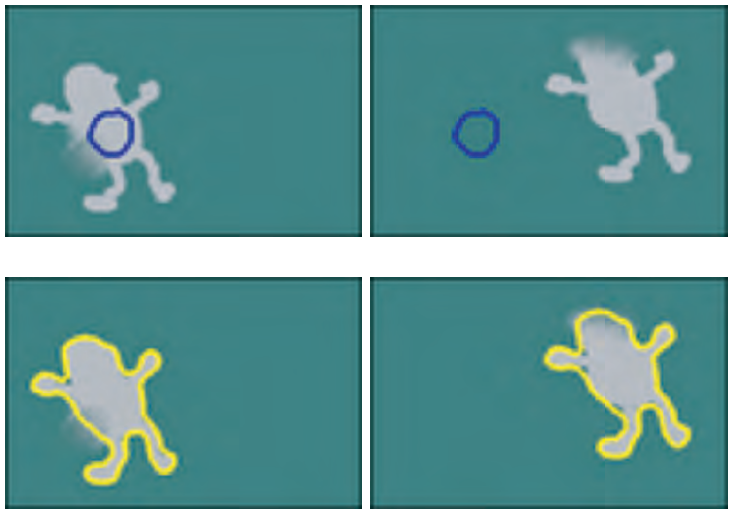


Fig. A.20. Logical OR model combines information. Initial contour (top) and final segmentation (bottom).



Fig. A.21. Typical behavior of the algorithm. Initial contour (top), end of initial registration phase (middle), and final segmentation (bottom).

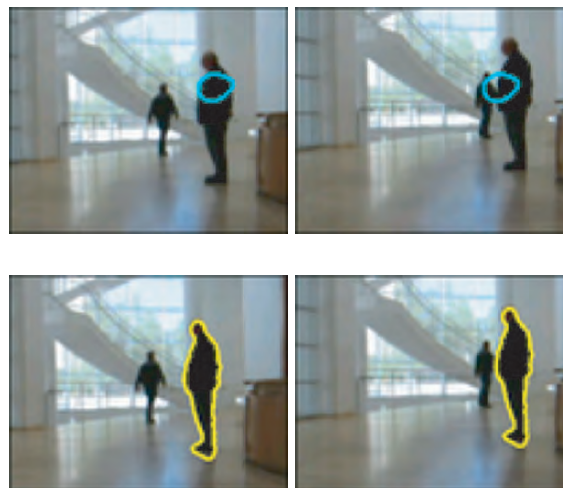


Fig. A.22. Logical AND model restricts the segmentation. Initial contour (top) and logical AND (bottom).